
BACHELORARBEIT

Frau
Michelle Albinus

**Untersuchungen zu Chancen
und Limits von Texture Maps
im Umfeld der Entwicklung di-
gitaler 3D Spiele**

2017

BACHELORARBEIT

Untersuchungen zu Chancen und Limits von Texture Maps im Umfeld der Entwicklung digitaler 3D Spiele

Autorin:
Frau Michelle Albinus

Studiengang:
Medieninformatik und interaktives Entertainment

Seminargruppe:
MI14w3-B

Erstprüfer:
Herr Prof. Dr.-Ing. Wilfried Schubert

Zweitprüfer:
Frau M.Sc. Rika Fleck

Einreichung:
Mittweida, 18. Dezember 2017

BACHELOR THESIS

Examining the Capabilities and Limitations of Texture Maps in Digital 3D Games Development

author:

Ms. Michelle Albinus

course of studies:

Media Informatics and Interactive Entertainment

seminar group:

MI14w3-B

first examiner:

Prof. Dr.-Ing- Wilfried Schubert

second examiner:

M.Sc. Rika Fleck

submission:

Mittweida, December 18 2017

Bibliografische Angaben

Nachname, Vorname :Albinus, Michelle:

Thema der Bachelorarbeit: Untersuchungen zu Chancen und Limits von Texture Maps im Umfeld der Entwicklung digitaler Spiele

Topic of thesis: Examining the Capabilities and Limitations of Texture Maps in Digital Games Development

95 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2017

Abstract

Der Fortschritt in Computerhard- und software ermöglicht der Spieleindustrie, immer technisch versiertere digitale Spiele zu entwickeln. Besonders der Bereich des 3D und 2D Designs profitiert davon, obgleich immer noch Grenzen vorzufinden sind, welche in Hinsicht auf die Generierung von Game Assets zu berücksichtigen sind. Die vorliegende Arbeit untersucht dabei den aktuellen Forschungsstand und analysiert diesbezüglich Chancen und Limits sogenannter Texture Maps zur Erzeugung von Detail auf dreidimensionalen Objekten. Die Resultate dieser Untersuchung sollen am ausgewählten Beispiel der Videospielreihe *Tomb Raider* aufgezeigt werden.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	V
Glossar.....	VII
Abkürzungsverzeichnis.....	XIV
Abbildungsverzeichnis.....	XV
Tabellenverzeichnis.....	XVII
1 Einleitung.....	1
2 Polygonales Modeling und Texturing.....	3
2.1 Definition Polygonales Modeling.....	3
2.2 Definition Texturing.....	4
3 Texture Mapping.....	5
3.1 Texture Mapping im Entwicklungsprozess von Game Assets.....	5
3.1.1 Texture Mapping im Kontext Grafiksysteme.....	6
3.1.2 Voraussetzungen.....	7
3.2 Arten von Texture Mapping.....	9
3.2.1 Diffuse/Albedo Mapping.....	9
3.2.2 Specular Mapping.....	10
3.2.3 Roughness Mapping.....	10
3.2.4 Occlusion Mapping.....	10
3.2.5 Normal/Bump Mapping.....	11
3.2.6 Displacement/Parallax Mapping.....	12
4 Chancen und Limits von Texture Mapping.....	16
4.1 Technischer Ansatz.....	16
4.1.1 Texture Map und 3D Model.....	16
4.1.2 Priorisierung von Game Assets	17
4.1.3 Kosten-Nutzen-Entscheidungen bei Surface-Detail Maps	18
4.2 Narrativer Ansatz.....	22
4.2.1 Narration durch Farbe.....	22
4.2.2 Narration durch Licht und Schatten.....	27

4.2.3	Narration durch Oberflächendetail.....	29
4.3	Realismus und Stilismus.....	31
4.3.1	Realismus durch Texturen.....	32
4.3.2	Stilismus durch Texturen.....	34
4.3.3	Stilisierter Realismus aus Texturen und Modeling-Stil.....	36
4.4	Procedural Textures.....	39
4.4.1	Definition und Wirkungsumfeld von procedural Textures.....	40
4.4.2	Potenzial von Procedural Textures.....	40
5	Analyse des Texture Mappings am Beispiel von <i>Tomb Raider's</i> Lara Croft.....	45
5.1	Vorstellung der Spielereihe.....	45
5.2	Charakteristik des Models „Lara Croft“.....	45
5.2.1	„Original“ Zeitachse.....	46
5.2.2	„Legend“ Zeitachse.....	46
5.2.3	„Survivor“ Zeitachse.....	47
5.3	Analyse des Texture Mappings.....	48
5.2.1	Texturing in der „Original“ Zeitachse.....	48
5.2.2	Texturing in der „Legend“ Zeitachse.....	52
5.2.3	Texturing in der „Survivor“ Zeitachse.....	54
5.4	Fazit zum Texture Mapping im <i>Tomb Raider</i> Franchise.....	59
6	Fazit.....	61
6.1	Zusammenfassende Schlussfolgerungen.....	61
6.2	Ausblick der Arbeit und weiterführende Gedanken.....	63
	Quellenverzeichnis.....	XVIII
	Literaturquellen.....	XVIII
	Internetquellen.....	XX
	Bildquellen.....	XXVII
	Eigenständigkeitserklärung.....	XXXI

Glossar

Adobe	Software-Firma für Entwicklung und Vertrieb von Kreativapplikationen, Anwendungen zur Digitalisierung von Dokumenten sowie Marketing-Lösungen ¹
ATI	Hersteller für Grafikkarten und Chipsätze (formals ATI, heute AMD) ²
Art-based	2D und/oder 3D Design betreffend
Artefakte	verzerrte Texel ³
Bitmap	digitale Bilddatei, gemessen in Pixel ⁴
Box	digitaler, dreidimensionaler Würfel
Box-Modeling	Modeling Methode, bei der aus primitiven Grundformen durch Transformation der Polygone ein komplexes 3D Modell geschaffen wird ⁵
Brightness	Helligkeit
Cel Shading	Darstellungsmethode-Methode, bei der 3D Objekte wie 2D Modelle gerendert werden ⁶
Code	Instruktionen in einer ausgewählten Programmiersprache ⁷
Core Design	britisches Game Development Studio ⁸
Crystal Dynamics	US-amerikanisches Game Development Studio ⁹
Cube	siehe <i>Box</i>

¹ Adobe Systems Software Ireland Ltd. (2017)[B]

² Archlinux (2017)

³ Tatarchuk (2005), S. 19

⁴ TechTerms (2014)

⁵ Gantzler (2005), S. 255

⁶ GIANT BOMB (2017)

⁷ WhatIs.com (2005)

⁸ Core-design.com (o.J.)

⁹ Crystald.com (o.J.)

Cylinder	digitaler, dreidimensionaler Zylinder
Edge	Kante eines Polygons
Edge Loops	Kantenringe um ein oder mehrere Polygone
Extrudieren	Transformationsoperation in einem 3D Modeling Programm zum Verschieben von Vertices, Edges oder Faces mit gleichzeitiger Topologie-Neubildung ¹⁰
Face	Fläche eines Polygons
Face Normal	orthogonaler Vektor auf einem Face ¹¹
Fragment Shader	programmierbarer Abschnitt in der Grafik-Pipeline, auf der Instruktionen für Pixel ausgeführt werden ¹²
Frame	gerenderte 2D Bild einer (3D) Szene ¹³
Frame Rate	Größe in der Computergrafik, die angibt, wie oft ein Frame in der Sekunde aktualisiert wird; gemessen in fps (frames per second) ¹⁴
Franchise	kooperatives Vertriebssystem, bei dem der Franchisenehmer gegen Gebühr ein vom Franchisegeber begründetes Geschäftskonzept nutzen darf ¹⁵
Game Asset	Objekt, welches in einem Videospiel verwendet wird ¹⁶
Game Development Studio	Unternehmen für die Entwicklung von Videospielen
Game Engine	Software-Plattform für Videospiele ¹⁷

¹⁰ Watkins (2001), S. 67

¹¹ Autodesk Inc. (2016)[B]

¹² Recker (2006), S. 10

¹³ Lee (2016), S. 110

¹⁴ Klappenbach (2017)

¹⁵ Deutscher Franchiseverband (2017)

¹⁶ Pettit (2015)[A]

Gameplay	Erfahrung des Spielers beim Spielen eines Videospiels durch Interaktion mit den Inhalten des Spiels und agieren innerhalb der Mechaniken und Regeln des Spiels ¹⁸
Gaming	Spielen von Videospielen
Geometrie	Polygonale Oberfläche eines 3D Modells
GIMP	kostenloses Programm zur Erzeugung und Manipulation von Pixelgrafiken ¹⁹
Hard Shading	Simulation von fehlender Kantentiefe durch Manipulation des Normalenwinkels auf null Grad ²⁰
Hero Asset	3D Modell, welches für eine für dieses Objekt einzigartige Textur gemappt wird (UV Raum wird von 0 bis 1 ausgenutzt, keine überstehenden UV Shells)
High-Poly	siehe <i>High-Resolution</i>
High Resolution	hoher polygonaler Detailgrad ²¹
Hue	Farbton
Layer	Ebene
Low-Poly	geringer polygonaler Detailgrad ²²
Material	Instruktionen im Render-Prozess zur Definition der Parameter für die Darstellung einzelner Texture Maps im Zusammenhang ²³
Mesh	digitales, dreidimensionales Netz aus Polygonen ²⁴

¹⁷ Gantzler (2005), S. 31

¹⁸ Techopedia (2017)

¹⁹ GIMP(o.J.)

²⁰ Autodesk Inc. (2015)[B]

²¹ Slick (2016)

²² Autodesk Inc. (2016)[B]

²³ Unity Technologies (2015)[B]

²⁴ Autodesk Inc. (2015)[A]

Model	digitales, dreidimensionales Modell
Nvidia	Unternehmen für Herstellung und Vertrieb von Grafikprozessoren und Chipsätzen ²⁵
Oversampling	ungenaueres Abtasten beim Ray-Tracing, wobei Schnittpunkte von Texture Map und Geometrie verpasst werden und somit Lücken beim Rendern entstehen ²⁶
Parallaxe	scheinbare Verschiebung der sichtbaren Geometrie aufgrund des Blickwinkels des Betrachters ²⁷
Performance	siehe <i>Frame Rate</i>
Photoshop	Produkt von Adobe; Software zur Erzeugung und Manipulation von Pixelgrafiken ²⁸
Physically-based Shading	Zusammenschluss von mehreren Render-Techniken, um eine physisch korrekte Imitation von Licht darzustellen ²⁹
Plug-In	zusätzlich in eine Software integriertes Teilprogramm
Polycount	Anzahl der Polygone in einer Szene ³⁰
Polygon	digital dargestellte, geschlossene Form, die von mindestens drei Vertices begrenzt ist ³¹
Preset	vorgefertigte, zugriffsbereite Ansammlung von Parametern
Procedural Texture	mathematisch generierte Texture Map ³²

²⁵ Nvidia (2017)

²⁶ Donnelly (2005), S. 127

²⁷ Tatarchuk (2005), S. 5

²⁸ Adobe Systems Software Ireland Ltd. (2017)[A]

²⁹ Learn OpenGL (o.J.)[B]

³⁰ Watkins (2001), S. 50

³¹ Gantzler (2005), S. 4

³² Pettit (2015)[A]

Quads	Polygone mit vier Vertices
Ray-Tracer	Algorithmus, der Lichtstrahlen umgekehrt (vom Betrachter zum Objekt) simuliert und so nach Schnittpunkten einer Texture Map mit polygonaler Geometrie sucht ³³
Renderer	Programmeinheit in der Grafik-Pipeline, wie eine dreidimensionale Szene als zweidimensionales Bild auf dem Monitor darstellt ³⁴
Rocksteady	Game Development Studio ³⁵
Sample Points	neu gesetzte Vertices eines Meshes oder Texel einer Texture Map im Displacement Mapping ³⁶
Saturation	Sättigung
Seams	Schnittkanten
Shader	Software- und Hardwareeinheiten, über die bestimmt wird, wie ein dreidimensionales Objekt gerendert werden soll
Shooter	Videospielgenre, bei der der Spieler Schusswaffen zur Bekämpfung von Gegnern nutzt
Soft Shading	Simulation von Kantentiefe durch Manipulation des Normalenwinkels auf 180 Grad ³⁷
Standard Texture Mapping	Texture Mapping verwendet Bitmap Bilddateien
Sphere	digitale, dreidimensionale Kugel
Sphere-Tracer	Rendering-Methode zur Lokalisierung von Schnittpunkten zwischen Texture Map und Geometrie mit

³³ Scratchapixel 2.0 (o.J.)

³⁴ Lee (2016), S. 110

³⁵ Rocksteady Ltd. (2017)

³⁶ Szirmay-Kalos, Umenhoffer (2006), S. 1

³⁷ Autodesk Inc. (2015)[C]

	Hilfe des Modells einer Kugel und der Verwendung des geometrischen Abstands ³⁸
Standard Asset	3D Modell, welches für die Verwendung einer tileable Texture gemappt wird (UV Raum wird über die Grenzen von 0 bis 1 genutzt)
Surface-Detail	Oberflächendetail eines digitalen, dreidimensionalen Modells
Tessellation	stärkere Unterteilung eines Meshes in Triangles ³⁹
Tessellation Shader	Abschnitt in der Grafik-Pipeline, der die Tessellationsparameter festlegt und wo die Tessellation durchgeführt wird ⁴⁰
Texel	ein Pixel in einer Textur ⁴¹
Tileable Texture	beliebig oft wiederholbare Texture Maps ohne sichtbaren Übergang an den Seams ⁴²
Topologie	der Oberfläche des 3D Modells strukturgebendes Netz aus Edges und Vertices ⁴³
Triangles	Polygone mit drei Vertices
Uncanny Valley	Theorie von Masahiro Mori, dass, je näher menschliche Reproduktionen echten Menschen kommen, desto mehr Missstände und feine Fehler fallen dem Betrachter auf ⁴⁴
User	Spieler von Videospielen
Vertex	Eckpunkt eines Polygons

³⁸ Recker (2006), S. 29

³⁹ Bailey; Cunningham (2016), S. 315

⁴⁰ Ebd., S. 316

⁴¹ Learn OpenGL (o.J.)[A]

⁴² Gantzler (2005), S. 30

⁴³ Polycount (2017)[B]

⁴⁴ Stuart (2015)

Vertex Shader	programmierbarer Abschnitt in der Grafik-Pipeline, auf dem Vertex-Transformationen ablaufen ⁴⁵
Vertices	Eckpunkte eines Polygons
Workflow	zirkulierendes Modell aus sich gegenseitig bedingenden Arbeitsschritten, um ein bestimmtes Ziel zu erreichen

⁴⁵ Recker (2006), S. 9

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
GPU	Graphics Processing Unit ⁴⁶
HSB	Hue, Saturation, Brightness ⁴⁷
PBR	Physically-based Shading ⁴⁸
PC	Personal Computer
pl.	Plural
PSP	PlayStation Portable ⁴⁹
RGB	Rot, Grün, Blau
Sci-Fi	Science Fiction

⁴⁶ GIGA (2016)

⁴⁷ Ahearn (2006); S. 14

⁴⁸ Seymour (2013)

⁴⁹ PlayStation (2017)

Abbildungsverzeichnis

Abbildung 1: grobes Schema zu ausgewählten Arbeitsschritten in der Rendering-Pipeline	7
Abbildung 2: Ausführungen einer Säule mit verschiedenem Polycount	17
Abbildung 3: Zustände des "Big Daddy" in BioShock	23
Abbildung 4: ADAM Injektor in BioShock.....	24
Abbildung 5: Farbschemata durch Texture Maps in Batman: Arkham City	25
Abbildung 6: Lagerfeuer in Horizon Zero Dawn, Dark Souls und Tomb Raider.....	26
Abbildung 7: Unverändertes und manipuliertes Startareal in Borderlands 2: Tiny Tina's Assault on Dragon Keep	27
Abbildung 8: Holzboden im Croft Anwesen in Rise of the Tomb Raider.....	28
Abbildung 9: Evolution von Nathan Drake in Uncharted 1-4	30
Abbildung 10: Falten in der Kleidung als Zeugnis tropischen Klimas in Uncharted 4: A Thief's End	31
Abbildung 11: Einwirken der Umgebung auf das Mesh unter Verwendung von PBR...	33
Abbildung 12: Erstellung der Diffuse Map eines Game Assets aus Borderlands	35
Abbildung 13: Wechsel von realistischer Grafik zu Comicstil in Borderlands	36
Abbildung 14: Stilisierte Texturen auf realistischer Modeling Grundlage in Life is Strange	38
Abbildung 15: Realistische Texturen auf Basis stilisiertem Modelings in Batman: Arkham Asylum	39
Abbildung 16: Viewport in Allegorithmic Substance Painter.....	43
Abbildung 17: Zusammenschnitt von Referenzbildern zu Tomb Raider: Festuring Lara Croft.....	48
Abbildung 18: Zusammenschnitt von Referenzbildern zu Tomb Raider II: Starring Lara Croft und Tomb Raider III: Adventures of Lara Croft.....	49

Abbildung 19: Zusammenschnitt von Referenzbildern zu Tomb Raider: The Last Revelation und Tomb Raider: Chronicles	50
Abbildung 20: Zusammenschnitt von Referenzbildern zu Tomb Raider: Angel of Darkness.....	51
Abbildung 21: Zusammenschnitt von Referenzbildern zu Tomb Raider: Legend	52
Abbildung 22: Zusammenschnitt von Referenzbildern zu Tomb Raider: Anniversary ..	53
Abbildung 23: Zusammenschnitt von Referenzbildern zu Tomb Raider: Underworld...	54
Abbildung 24: Zusammenschnitt von Referenzbildern zu Tomb Raider (2013).....	56
Abbildung 25: Referenzbilder zu Lara Croft in Rise of the Tomb Raider	58
Abbildung 26: Die verschiedenen Lara Crofts im Tomb Raider Franchise	59

Tabellenverzeichnis

Tabelle 1: Gegenüberstellung von Surface-Detail Mapping Methoden hinsichtlich Funktion und Anwendung.....	14
---	----

1 Einleitung

Videospiele erfreuen sich einer immer größeren Popularität. Wo 2016 weltweit 2.099 Millionen Menschen Zeit mit Gaming verbrachten⁵⁰, sind 2017 bereits 2200 Millionen zu erwarten⁵¹. Die Gemeinschaft der Gamer nimmt demnach zu, obgleich der Gegenstand ihres Hobbies vielfältig in Erscheinung tritt. Gespielt wird heutzutage nicht nur auf dem Computer; auch Heimkonsolen, Tablets, Smartphones und Browser treten als Bühne für Videospiele in Erscheinung.⁵² Die Entwicklung der dort vertretenen virtuellen Welten ist dabei die Aufgabe eines gesamten Teams, die in sogenannten Game Development Studios designen und programmieren, um letztendlich ein digital interaktives Gesamtwerk zu erschaffen.

So wie sich die Angebotsbreite der Plattformen erweitert hat, erreichen Videospiele besonders hinsichtlich ihrer technischen sowie grafischen Reife einen neuen Horizont.⁵³ Laut Spielewissenschaftler Richard Huddy kann die Gaming-Gemeinschaft innerhalb von fünf Jahren eine Entwicklung der grafischen Verfügbarkeiten bis hin zum Fotorealismus erwarten.⁵⁴ Durch welche Techniken dieser Zustand zu ermöglichen sein wird, bleibt allerdings noch aus. Währenddessen bedienen sich Designer vor allem der Methode des *Texturings*, um einen hohen Detailgrad der Objekte, welche als *Game Assets* in digitalen Spielen verwendet werden, zu gewährleisten.⁵⁵ Im Gegensatz zu dreidimensionalen Modellen in Filmen sind jene in Videospielen durch die Anzahl ihrer Polygone limitiert. Für den Designer bedeutet das folglich, dass das eigentliche 3D Objekt nur eine recht geringe Auflösung aufweisen darf, während verschiedene *Texturen* durch ihre hohe grafische Qualität und Oberflächenbestimmende Eigenschaften dem Modell scheinbare Details auftragen. Durch welche Aspekte, Verfahren und Zusammenhänge mit den verschiedenen Bereichen der Game Asset Erzeugung zeigt sich nun jedoch das Potenzial von Texturen Maps bei der Entwicklung digitaler Videospiele und inwiefern stoßen sie an technische und stilistische Grenzen? Dieser Forschungsfrage soll in der vorliegenden Arbeit untersucht werden.

Welche Arten des Texture Mappings derzeit am häufigsten zum Einsatz kommen und mit welchen Resultaten sie wie funktionieren, soll zunächst als Basis geklärt werden. Hierbei soll nicht die Gesamtheit aller derzeit existierenden Texturing-Verfahren aufgezeigt werden, sondern eine Auswahl an verschiedenen Ansätzen erfolgen, um das Spektrum auf einem übersichtlichen Niveau zu halten. Da aber die Voraussetzungen,

⁵⁰ Newzoo (2016)

⁵¹ Newzoo (2017)

⁵² Ebd.

⁵³ Tavinor (2009), S. 1

⁵⁴ Sacco (2014)

⁵⁵ Pluralsight (2014)

Verfahrensanwendungen und das Wirkungsumfeld von Texturen ebenfalls durch weitere Segmente der Videospielentwicklung beeinflusst werden, müssen diese Zusammenhänge zusätzlich in diesem ersten deskriptiven Teil definiert werden. An dieser Stelle liegt der Fokus besonders auf der Position sowie Wirkungsweise des Texturings im Prozess der Game Asset Produktion. In einem zweiten Teil werden aufbauend auf dem vorherigen Abschnitt Chancen und Grenzen von Texture Maps anhand eines technischen, narrativen und stilistisch-realistischen Ansatzes erörtert werden, sodass sie anschließend im dritten und letzten Teil auf ein Beispiel angewendet werden können. Für diesen Zweck wird die Videospielreihe *Tomb Raider* herangezogen; die daran ausgeführte Analyse des verwendeten Texture Mappings soll dabei von der langjährigen Existenz des Franchises und dessen technischen Evolution profitieren. Die drei vorgestellten Teilabschnitte bauen hierbei aufeinander auf und bilden die Argumentationskette für ein Fazit, welches nicht nur die Zusammenfassung der Ergebnisse repräsentieren soll, sondern im gleichen Zuge eine Empfehlung für die Handhabung von Texturen darstellt.

Diese Arbeit ist vor allem aus designtechnischer Perspektive zu sehen; sie dient der Vermittlung der für einen Texture Designer relevanten Aspekte der Gestaltung durch Texturen, sodass deshalb keine tiefgreifenden programmiertechnischen Erläuterungen erwartet werden dürfen. Der wissenschaftliche Anspruch der Thematik erlaubt jedoch kurze Exkurse in ausgewählte technische Funktionsweisen hinter dem Wirkungsumfeld des Texturings, um die vorgestellten Zusammenhänge verstehen zu können.

2 Polygonales Modeling und Texturing

Bevor sich an die Definition des Texturings gewagt werden kann, muss als ein erster Schritt das polygonale Modeling verstanden werden, da dies nicht nur die Basis für die Generierung von Texture Maps darstellt, sondern zudem viele für das Texture Mapping essenzielle Begrifflichkeiten enthält.

2.1 Definition Polygonales Modeling

3D Modeling bezeichnet zunächst die digitale Darstellung von Objekten im dreidimensionalen Raum mittels Höhe, Breite und Tiefe.⁵⁶ Dabei wird eine Position in diesem Raum durch einen Punkt (*vertex*, pl. *vertices*) beschrieben. Die Verbindung zweier solcher Punkte bildet eine Linie oder auch Kante (*edge*), welche zwar eine Länge, jedoch in diesem Stadium noch keine Fläche (*face*) besitzt. Diese kommt erst dann zu Stande, wenn mindestens drei Linien einen Raum einschließen, wiederum durch mindestens dieselbe Anzahl an Eckpunkten verbunden. Diese geschlossene Form wird als *Polygon* betitelt.⁵⁷ Vereinen sich nun mehrere Polygone, entsteht schließlich ein dreidimensionales Objekt, welches eine Höhen-, Breiten- und Tiefeninformation aufweist.

Das Verfahren, durch welches ein solches Modell erzeugt wird, wird mit dem Begriff *polygonales Modeling* beschrieben.⁵⁸ Der Term indiziert außerdem, wie das Verfahren vonstattengeht; durch Verschieben, Rotieren und Skalieren der einzelnen Polygone wird das 3D Objekt Stück für Stück erzeugt. Diese können durch weitere mögliche Operationen manipuliert werden, obgleich nebst den bereits genannten drei Grundtransformationen das Extrudieren, Spiegeln und Unterteilen die gängigen Werkzeuge darstellen.⁵⁹ Hierzu entwickelt sich das angestrebte Zielmodell aus einfachen geometrischen Figuren (*primitives*), wie beispielsweise Würfel (*cube/box*), Kugel (*sphere*) oder Zylinder (*cylinder*), weswegen dieses Verfahren auch unter dem Begriff des *Box-Modelings* bekannt ist.⁶⁰ In Bezug auf die Erstellung von Game Assets wird die Zahl der Polygone, welche das finale Objekt bilden, oft auf einen bestimmten *Polycount* begrenzt. Dieser gibt an, wie viele einzelne Faces das Model aufweisen darf, um verwendet zu werden.⁶¹ All diese Bausteine bilden nun die Grundlage, auf dessen Basis das eigentliche Texturing erklärt werden kann.

⁵⁶ Patnode (2008), S. 2

⁵⁷ Gantzler (2005), S. 4

⁵⁸ Plowman (2016), S. 47

⁵⁹ Ebd., S. 46-57

⁶⁰ Gantzler (2005), S. 80

⁶¹ Watkins (2011), S. 50

2.2 Definition Texturing

Texturing umreißt den Prozess in der Generierung von *Art-based Game Assets*, bei welchem das dreidimensionale Modell eine Oberfläche mit spezifischen Merkmalen erhält.⁶² Bereits im Modeling-Prozess werden erste Überlegungen zum Texturing angestellt, um ausgehend davon Entscheidungen hinsichtlich der Geometrie zu treffen. Zu wissen, welche Materialien das Objekt verwendet und an welchen Stellen diese sich voneinander abgrenzen, kann dabei die Geometrie des 3D Modells bestimmen. Bereiche, in denen ein Oberflächenwechsel stattfindet, können somit frühzeitig im Asset *Workflow* berücksichtigt werden, um so das Texturieren zu erleichtern.

Um den Prozess des Texturings zu verstehen, ist es zunächst unumgänglich zu klären, was der Term Textur (Texture Map) bedeutet. Texturen sind als Pixelgrafik beschriebene *Bitmap* Bilddateien, welche Auskunft über das Material des Modells geben, indem sie auf dessen Oberfläche angezeigt werden. Das Anbringen solch einer Textur auf das dreidimensionale Objekt wird schließlich als *Texture Mapping* bezeichnet. Texturen können in jeglichem Grafikbearbeitungsprogramm, wie beispielsweise *GIMP* oder *Photoshop*, erstellt und bearbeitet werden.

Die Applikation einer Texture Map erfolgt dabei jedoch als dualistischer Prozess. In einem ersten Schritt muss dem Mesh mitgeteilt werden, wie es die visuelle Information aufnehmen soll. Inwiefern die Bilddatei sich um das Modell legen soll, wird mittels der UV Koordinaten beim sogenannten *UV Mapping* festgelegt.⁶³ Dabei repräsentieren die UV Koordinaten im Gegensatz zu XYZ keine Punkte im 3D Raum, sondern Positionen in der 2D Bilddatei.⁶⁴ Daher verlangt dieser Verfahren das Auffalten des *Mesches* entlang seiner Kanten zu einer zweidimensionalen Repräsentation seiner selbst, um das dadurch entstandene Netz als Vorgabe für die Generierung der Textur verwenden zu können. In einem zweiten Schritt – vorausgesetzt die Textur wurde nun nach dieser Vorlage erstellt – ergibt sich die Notwendigkeit, die Texture Map dem 3D Modell zuzuweisen. Integriert wird sie dabei in ein *Material*, welches alle Texturen in sich vereint.⁶⁵

⁶² Ebd., S. 28

⁶³ Plowman (2016), S. 58

⁶⁴ Gantzler (2005), S. 87

⁶⁵ Unity Technologies (2017)[B]

3 Texture Mapping

Bereits in diesen ersten Abschnitten wird deutlich, dass Texture Maps an gewisse Vorleistungen anderer Bereiche der Spieleentwicklung gekoppelt sind. Aus diesem Grund muss zunächst der Kontext begriffen werden, in welchem sich Texture Maps bewegen, bevor eine tiefgreifendere Untersuchung der Mapping Verfahren selbst vorgenommen werden kann. Neben ihren Einsatzmöglichkeiten müssen vor allem die technische Wirkungsweise dieser Methoden sowie ihre Voraussetzungen beleuchtet werden, ohne die eine Analyse des Potenzials von Texture Maps nicht möglich wäre.

3.1 Texture Mapping im Entwicklungsprozess von Game Assets

Die Prozesse des Modelings und Texturings finden nicht nur in der Spieleindustrie Anwendung; in der Filmbranche oder auch zu Werbezwecken werden heutzutage dreidimensionale Objekte zum Einsatz gebracht. Diese unterscheiden sich jedoch maßgeblich von 3D Modellen wie sie – als sogenannte Game Assets – in Videospielen auftreten.⁶⁶ Assets als Oberbegriff meint hierbei von der *Game Engine* verwendbare Dateien aus verschiedenen Bereichen wie beispielsweise Ton, Animation, 2D oder 3D Design. Sie bilden Inhalt und Substanz des Spiels und werden über die entsprechenden Ausgabegeräte dem *User* präsentiert.⁶⁷ Für die Untersuchungen dieser Arbeit liegt der Fokus jedoch ausschließlich auf den Grafik-basierten Assets. Als Art Assets werden jegliche Assets verstanden, welche der visuellen Darstellung dienen. Neben Licht und Animationen erweisen sich besonders dreidimensionale Modelle sowie Texturen als essenziell.

Insofern, dass Texturen in Pixelgrafiken dargestellt werden, sind auch sie zweidimensionale Bilder, welche durch Länge und Breite definiert werden. Sie manipulieren die Oberfläche des 3D Modelles in den meisten Fällen nur scheinbar, beeinflussen jedoch nicht die Topologie des Meshes. Gemessen werden sie hierbei in *Texel*, welches auf einen Pixel in der Textur verweist und genau eine Farbinformation speichert. Das gesamte Bild wird infolgedessen aus einer gitterartigen Ansammlung mehrerer Texel gebildet. Die Anzahl der Pixel in diesem Netz gibt zudem Auskunft über die Auflösung der Bilddatei. Je mehr Pixel verwendet werden, desto höher ist die Auflösung der Textur.⁶⁸

⁶⁶ Pettit (2015)[A]

⁶⁷ Davis (2009)

⁶⁸ Thorn (2013), S. 38

Bei der Generierung der eigentlichen Texture Maps obliegt es dem Designer nun, zu entscheiden, welche Eigenschaften die Oberfläche des Objektes besitzen soll und damit auch, welche Texture Maps angefertigt werden müssen. Dabei muss er im Stande sein, Materialien aus der Realität digital nachbilden zu können. Diesbezüglich erfordert es die Fähigkeit, sich das gewünschte Material auf dem 3D Modell vorzustellen, um es anschließend in seine Bestandteile zu zerlegen.⁶⁹ So wie die Oberfläche eines jeden Objektes in der Wirklichkeit verschiedene Merkmale aufweist, existieren ebenfalls unterschiedliche Texturtypen. Ein Gegenstand kann dabei grundlegend durch seine Färbung beschrieben werden, obgleich seine reale Beschaffenheit durch weitere Faktoren definiert wird. Einerseits besitzen dreidimensionale Objekte Höhen- und Tiefeninformationen, welche ihrer Oberfläche zunächst strukturelles Detail ermöglichen. Des Weiteren können diese jedoch nur wahrgenommen werden, wenn der Körper auf Licht reagiert. Wie genau sich demnach realitätsgetreues Detail in Form von Erhöhungen, Reflexionen, Absorptionen und Schatten ausbilden, wird durch mehrere Texturen ausgedrückt.⁷⁰

3.1.1 Texture Mapping im Kontext Grafiksysteme

Um im Folgenden die verschiedenen Mapping Methoden nachvollziehen zu können, erfordert es einen Exkurs in die Funktions- und Arbeitsweise des Texture Mappings.

Jedes Game Asset, dass im Endprodukt auf dem Bildschirm zu sehen sein soll, muss zunächst gerendert werden. Selbst wenn das Videospiel in 3D Grafik generiert wird, handelt es sich bei dem auf dem Bildschirm sichtbaren Resultat um ein zweidimensionales Bild (*Frame*). Die Darstellung einer Szene aus zwei- oder dreidimensionalen Informationen mittels *GPU* wird als *Rendering* bezeichnet.⁷¹ Während beispielsweise bei Filmen oder digitalen Bildern das *Software Rendering* zum Einsatz kommt, wird bei Videospielen auf das sogenannte *Hardware Rendering* zurückgegriffen. Hierbei muss die Grafikkarte des Computers oder der Konsole das entsprechende Bild in Echtzeit darstellen. Um ein flüssiges Spiel ausgeben zu können, müssen die Anzahl der Polygone in der Szene sowie die Auflösung der Texturen so gewählt sein, dass innerhalb einer Sekunde mindestens 30 Frames gerendert werden können.⁷²

⁶⁹ Ahearn (2016), S. 2

⁷⁰ De Vires (2015), S. 146

⁷¹ Lee (2016), S. 110

⁷² Watkins (2011), S. 12

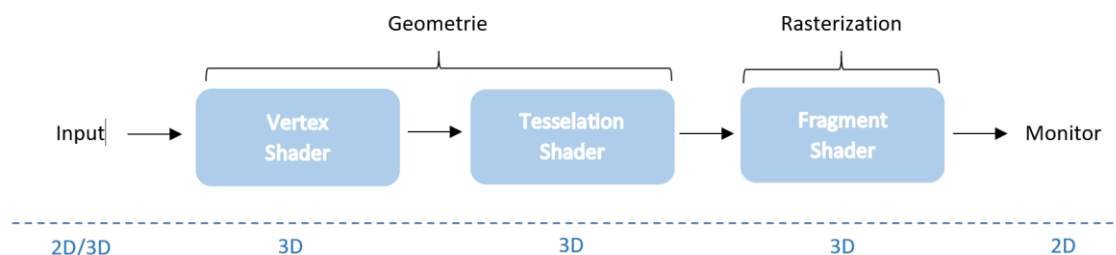


Abbildung 1: grobes Schema zu ausgewählten Arbeitsschritten in der Rendering-Pipeline

Die Rendering *Pipeline* besteht weitestgehend aus zwei großen Teilen; zuerst werden alle 3D Koordinaten in 2D Koordinaten umgewandelt und diese wiederum im Anschluss in Pixel mit einer Farbinformation. Dieses Verfahren kann jedoch nicht im Ganzen durchgeführt werden, sondern mit Hilfe mehrerer Unterprozesse, *Shader* genannt.⁷³ Nachdem die erforderliche Datenmenge an Polygonen in die Pipeline eingelesen wurde, agiert zunächst der *Vertex Shader*, indem er ausgehend von den Werten, die für jeden einzelnen Vertex gespeichert sind, wie beispielsweise Position, Farbe oder Texturkoordinaten, die entsprechenden Transformationen durchläuft. Durch zusätzliche Shader können im Anschluss verschiedene Manipulationen am Mesh vorgenommen werden, bis dieses in dreieckige Polygone (*Triangles*) unterteilt ist. Diese Fragmente dienen nun als Input für den *Fragment Shader*, in welchem sie einen Texturwert erhalten.⁷⁴ Die nötigen Kalkulationen für diesen Schritt werden dabei im Shader gespeichert, über welchen sich zudem die verschiedenen möglichen Variablen initiieren lassen. Diese werden anschließend in dem dem Shader zugewiesenen Material als Texturkanäle angezeigt, welchem nun die passenden Texture Maps zugeordnet werden. Das Material beinhaltet außerdem weitere Anweisungen, inwiefern die Oberfläche des 3D Objekts gerendert werden soll. Es können infolgedessen weitere Parameter justiert werden, beispielsweise wie oft die Textur auf dem Objekt wiederholt werden soll oder inwieweit mehrere Texturen ineinander überblenden.⁷⁵ Die hierbei ablaufenden Prozesse sollten dabei jedem Designer geläufig sein, da sie an einer späteren Stelle als Grundlage für das Kosten-Nutzen-Kalkül der einzelnen Mapping Methoden gebraucht werden.

3.1.2 Voraussetzungen

Texturen könne somit zwar als separate Elemente verstanden werden, sie sind jedoch nur in Kombination mit einem dreidimensionalen Objekt gerechtfertigt existent. Da insofern sich Modeling und Texturing gegenseitig bedingen, bestehen auch gewisse Faktoren, die es zu beachten gilt, wenn ein Game Asset gefertigt werden muss.

⁷³ De Vires (2015), S. 35

⁷⁴ Recker (2006), S. 9-10

⁷⁵ Unity Technologies (2017)[B]

Zunächst ist es nicht nur möglich, eine Textur passend zu einem 3D Objekt zu erstellen. Zum Teil werden Modelle auch ausgehend von der bereits bestehenden Texture Map – einer Referenz – generiert. Allerdings bevorzugen Designer, die mittels *Box-Modeling* arbeiten, ersteres, um eine bessere Kontrolle über die Arbeit am Mesh zu gewährleisten. Dieser Fall verlangt daher ein dreidimensionales Objekt als Ausgangspunkt des Texture Mappings. Um mit dem UV Mapping fortfahren zu können, muss dieses wiederum in einer geeigneten *Topologie*, das heißt in einem organisiert strukturiertem Polygonnetz, vorliegen. Das Mapping kann zwar mit Polygonen verschieden hoher Seitenzahl durchgeführt werden, bevorzugt wird jedoch das Arbeiten mit vierseitigen Polygonen (*Quads*).⁷⁶ Der Vorteil einer solchen Struktur besteht in der Aufteilung des Objektes in Reihen und Spalten, wodurch sich allumfassend *Edge Loops* im Modell finden lassen. Diese helfen anschließend beim Mapping der UV Koordinaten, da sie sich besonders an Schnittstellen verschiedener Bereiche des Modells, welche infolgedessen oft unterschiedliche Farbmerkmale aufweisen, als Schnittkanten (*Seams*) anbieten. N-Gone, Polygone mit mehr als vier Vertices, sollten dabei vor dem UV Mapping zu Quads oder Triangles verbunden werden. Unterteilungen, die nicht für das Mapping benötigt werden, sollten dabei so weit wie möglich aufgetrennt und zu größeren Quads oder Triangles aufgelöst werden, um Polygone zu sparen. So kann das Mesh sauber in mehrere *Texture Shells* unterteilt werden und optimal für das Texturing im zweidimensionalen Raum ausgelegt werden.⁷⁷

Weiterhin bestehen Restriktionen bezüglich der Auflösung der geplanten Texturen, wessen sich der Designer im Vorfeld bewusst sein sollte. Die meisten Grafikkarten und Game Engines können Texturen der Zweierpotenz schneller verarbeiten und abbilden, obgleich einige sogar den Darstellungsprozess abbrechen, da sie ihn ausschließlich mit solchen Maßen ausführen können. Die untere Grenze stellt dabei eine Abmessung von 16x16 Pixeln dar, wobei so geringe Auflösungsformate häufig nur für Partikeleffekte genutzt werden. Je nach Bedarf liegt die Standardgröße von Texturen beim *Standard Texture Mapping* derzeit bei 1024x1024 Pixel. Besonders für Charakter Game Assets oder andere Modelle, die beispielsweise häufig oder von Nahem durch den Spieler betrachtet werden und folglich besonders hoch aufgelöst sein müssen, kann die Größe der Texture Map weitaus höher liegen. Allerdings ist die Form der Textur dabei nicht auf ein Quadrat beschränkt, rechteckige Maße wie zum Beispiel 512x1024 Pixel sind ebenfalls durchaus üblich.⁷⁸

⁷⁶ Watkins (2011), S.15

⁷⁷ Gantzler (2005), S. 256

⁷⁸ Ahearn (2006), S. 54-55

3.2 Arten von Texture Mapping

Da ein Material mehrere Texturkanäle beinhalten kann, ist es demzufolge möglich, verschiedene Texturen auf ein und dasselbe Material zuzuordnen. Dadurch, dass unterschiedliche Engines diese Optionen verschieden umsetzen und infolgedessen die einzelnen nötigen Elemente auf andere Weise benennen, verschwimmt die Grenze zwischen Mapping und Texture Map oft. Die beiden Terme sind jedoch konsequent voneinander zu trennen, da das Texture Mapping die Instruktionen des Shaders darstellen, inwiefern die Texture Map auf dem Mesh gerendert werden soll. Das Mapping stellt demnach das Verfahren dar, während die Map den dazu nötige Datensatz repräsentiert.⁷⁹

3.2.1 Diffuse/Opacity Mapping

Die Diffuse oder auch Albedo Map beschreibt die grundlegende Farbe, die ein Modell aufweist. Ist eine solche Map dem zum Mesh gehörenden Material zugeordnet, wird dessen Standardfarbe ersetzt.⁸⁰ Die Textur verwendet dabei den RGB Kanal sowie das HSB Modell. Einerseits können der Bilddatei dadurch Farbinformationen entnommen werden, welche als Rot-, Grün- und Blauwerte von 0 bis 255 gespeichert werden. Andererseits werden diese zusätzlich manipuliert durch Sättigung, Helligkeit und Farbton, erstere als Werte von 0 bis 100 und letzteres von 0 bis 360.⁸¹ Als Farbton wird dabei die Farbfamilie bezeichnet, zu der die Farbe gehört. Indigo, Ultramarin und Cyan stellen beispielsweise Mitglieder der Blautöne dar. Je mehr Weißanteil eine Farbe aufweist, desto weniger gesättigt erscheint sie und je größer der Schwarzanteil, desto geringer ist ihre Helligkeit.⁸²

In manchen Fällen ist es zudem ratsam, blickdurchlässige Freiräume bei Game Assets durch Transparenz in der Textur darzustellen als durch unnötig viele Polygone. Beispielsweise wird nicht jedes Geflecht eines Maschendrahtzauns ausmodelliert, sondern nur der Umriss des gesamten Zauns. Um die Lücken zwischen den Drähten durchsichtig zu machen, wird entweder eine Opacity Map oder ein Alpha Kanal in der Diffuse Map erstellt. Nutzt der Designer eine eigenständige Texture Map, wird diese in einem extra Kanal im Material abgelegt, im Gegensatz dazu ist der Alpha Kanal direkt in der Diffuse Map gespeichert. Transparenz wird an dieser Stelle durch Graustufen erzeugt; weiß repräsentiert dabei volle Deckkraft der Diffuse Textur, während schwarz totale

⁷⁹ Balazs (2017)

⁸⁰ Hodges (2003), S. 260

⁸¹ Ahearn (2016), S. 15-17

⁸² Mollica (2013), S. 14-16

Transparenz generiert. Die dazwischen liegenden Grautöne bilden demnach die verschiedenen Grade von Transparenz ab.⁸³

3.2.2 Specular Mapping

Dieser Typus ist entscheidend für die direkte Lichtreflexion eines Objektes. Der Kanal des Materials für die Specular Map bedient sich hierzu der Graustufen der Bilddatei, um Position und Intensität von Glanzpunkten zu kontrollieren. Weiß repräsentiert dabei volle Reflexion, während schwarz Glanz komplett unterdrückt.⁸⁴ Zur Generierung solch einer Textur erfordert es meist nur die Diffuse Map, dessen Sättigung im Grafikbearbeitungsprogramm zunächst auf null gesetzt und anschließend mittels Farbwertmanipulationen bis zum gewünschten Resultat manipuliert werden kann.

3.2.3 Roughness Mapping

Zunächst scheint sich die *Roughness Map* nicht sonderlich von der Specular Map unterscheiden; sie repräsentiert zwar ebenfalls die Reflektivität eines Materials, beleuchtet aber einen anderen Aspekt. Während die Specular Map vermittelt, wie viel Licht zurückgeworfen wird, definiert die Roughness Map die Art und Weise, wie es abgegeben wird. Ein Objekt kann dabei entweder glänzend und klar, wie beispielsweise einige Metalle, oder stumpf, wie Holz oder Erde, sein.⁸⁵ Ein Goldklumpen kann beispielsweise nur wenig Licht reflektieren, seine Oberfläche gibt das wenige Licht jedoch trotz dessen in einem scharfen Winkel ab, was vor allem an einem kleinen Glanzpunkt mit harten Kanten erkennbar ist. Im Gegensatz dazu kann ein Stück Holz viel Licht reflektieren – wenn es zum Beispiel nass ist – jedoch wird das einfallende Licht dabei viel breiter auf dessen Oberfläche gestreut und wirkt dadurch unscharf und verwaschen. Dieser Glanzpunkt würde dabei einen größeren Durchmesser und weiche, weit auslaufende Kanten aufweisen. Auch an dieser Stelle kommt erneut eine Graustufen Textur zum Einsatz, wobei schwarz hundert Prozent stumpf und weiß hundert Prozent glänzend bedeuten.⁸⁶

3.2.4 Occlusion Mapping

So wie Reflexionen durch Lichtquellen das Modell direkt ausleuchten, affektiert auch indirektes Licht der Umgebung das Objekt. Teile eines Modells, welche durch überstehende Objekte verdeckt werden, sollen dabei nicht beleuchtet werden, sondern Schat-

⁸³ Gantzler (2005), S. 109-110, 116

⁸⁴ Hodges (2003), S. 260

⁸⁵ Seymour (2013)

⁸⁶ Ebd.

ten ausbilden. Um solch ein authentisches Feedback zu erzielen, ist es erforderlich, mit Hilfe einer Occlusion Map festzulegen, welche Bereiche des Meshes vom indirekten Lichteinfall betroffen sein sollen. Bei nur standardmäßig ausgeleuchteten Modellen bilden sich lediglich großflächige Schatten mit abrupten Rändern aus. Der Vorteil beim Occlusion Mapping ist im Gegensatz der, dass hier stärkere Kontraste erzeugt werden können, da weite Schattenkanten mit einer weichen Ausblendung generiert werden.⁸⁷ Auch diese Form der Textur nutzt Farbwerte von weiß bis schwarz, obgleich weiß volle Reflexion und schwarz das Ausbleiben dieser übermittelt. Neben der Möglichkeit die Texture Map manuell durch Grafikmanipulation herzustellen, ist es ebenfalls üblich, besonders bei komplexeren Objekten auf 3D Programme zurückzugreifen, in welchen die Occlusion Map direkt vom Mesh abgeleitet und extrahiert werden kann.⁸⁸

3.2.5 Normal/Bump Mapping

Normal sowie Bump Mapping sind Methoden, um scheinbar gemodeltes Detail auf die Oberfläche des Objekts zu projizieren. Dabei verwenden sie jedoch verschiedene Maps als Datensatz. Das Bump Mapping greift auf die sogenannte Height Map zurück, wobei hingegen das Normal Mapping die namensgebende Normal Map nutzt.⁸⁹

Normal Maps weisen ein bläulich, purpurnes Erscheinungsbild auf; diese charakteristische Färbung geht auf die Bezeichnung dieser Textur zurück und referiert somit auf die sogenannten Normalen (*face normals*). Jedes Polygon besitzt solche eine Normale in Form eines orthogonalen Vektors auf dessen Oberfläche. Diese zeigt zudem an, in welche Richtung das Face zeigt. Eine Normale wird hierbei durch die Anordnung der Eckpunkte, die das Polygon bilden, geschaffen. Sind die Punkte im Uhrzeigersinn arrangiert, zeigt die Normale nach unten und das Face ist nur von hinten sichtbar, während sie bei einer Punktreihenfolge gegen den Uhrzeigersinn nach oben weist und damit auch die Fläche nach vorn.⁹⁰ Dieser Sachverhalt ist essenziell, um zu verstehen, dass dadurch das Licht berechnet werden kann, welches auf das Polygon einfällt. Solange die Normale senkrecht vom Polygon strahlt, wird dieses als flach wahrgenommen. Verändert sich allerdings deren Ausrichtung zu einer der drei Achsen im 3D Raum hin, erscheinen Erhöhungen und Absenkungen. Die Normal Map ersetzt hierbei die bisherigen Flächennormalen des Meshes durch die, die sie in veränderter Achsenstellung gespeichert hat, welche nun in die Lichtneukalkulation eingeschlossen werden und auf diese Weise Detail erscheinen lassen, welches auf dem 3D Objekt wäre, wenn ein Designer es zuvor auf der Oberfläche modelliert hätte. Die Informationen aus der Bilddatei – Rot, Blau und Grün - repräsentieren dabei die Orientierung der Normalen

⁸⁷ Geforce (2017)

⁸⁸ Unity Technologies (2017)[C]

⁸⁹ Balazs (2017)

⁹⁰ Autodesk Inc. (2016)[B]

nach den Achsen X (rot), Y (grün) sowie Z (blau). Die eigentliche Form des Modells wird durch diese Art des Mappings nicht physisch manipuliert.

Normal Maps können zum einen direkt aus der Diffuse Map – beispielsweise über *Plug-ins* oder entsprechende Online-Applikationen – hergeleitet werden. Zum anderen ist es besonders bei komplexen Objekten mit vielen Details effektiv, die Normal Map von einem *High-Resolution* Mesh auf ein *Low-Poly* Mesh zu *baken*. Dazu müssen zwei separate Modelle erstellt werden; einerseits das Mesh, wie es später im Spiel verwendet wird, und andererseits das gleiche Modell mit einer um ein Vielfaches höheren Auflösung und Detailstufe. Letzteres wird als Quelle für die Information über die Normalen ausgewählt, aus welcher nun die Normal Map extrahiert und schlussendlich auf das 3D Objekt mit der geringeren Auflösung projiziert werden kann. Wird das Game Asset nun im Spiel gerendert, berechnet die Engine das Licht anhand der Normalen des *High-Poly* Meshes und spiegelt die daraus resultierenden Erhebungen und Dellen auf dem Low-Poly Mesh wider.⁹¹

Height Maps, welche mitunter beim Bump Mapping zum Einsatz kommen, hingegen stellen eine Bilddatei in Graustufen dar, obgleich schwarz die tiefste Position eines Pixels und weiß die höchste repräsentiert.⁹² Während bei der Normal Map die Flächennormalen bereits in der Textur gespeichert und direkt an den Renderer übergeben werden kann, müssen diese erst noch anhand der Height Map aus den ursprünglichen Normalen der vorhandenen Polygone rekonstruiert werden.⁹³

Bei beiden Mapping Verfahren verformen die Geometrie des 3D Modells nicht, weswegen auch bei einem geringen Polycount das Mesh einen erhöhten Detailgrad aufweisen kann. Allerdings entsteht auf diese Weise auch keine der Map entsprechende Silhouette, wodurch die Textur bei steilen Blickwinkeln auf das Objekt verzerrt erscheinen kann.⁹⁴ Aus diesem Grund können mittels dieser Verfahren ebenfalls nur geringfügig Tiefeninformationen erzeugt werden, welche zwar die Oberfläche eines Materials spezifizieren – beispielsweise wie die Maserung von Holz oder Kratzer im Metall – ein Objekt jedoch nicht volumetrisch darstellen können.

3.2.6 Displacement/Parallax Mapping

Displacement Mapping funktioniert, indem der Mapping Algorithmus die gespeicherten Informationen zu den Flächennormalen einer Height Map abfragt, um die bisherigen Normalen des Meshes durch sogenannte *Sample Points* zu ersetzen. Letztere können dabei entweder Vertices oder Pixel darstellen, welche jeweils in einem anderen Sha-

⁹¹ Pettit (2015)[B]

⁹² Unity Technologies (2017)[A]

⁹³ Balazs (2017)

⁹⁴ Tartachuk (2012), S. 5

der-Typus verarbeitet werden. Vertices werden zunächst vom *Vertex Shader* behandelt, wonach diese Methode Per-Vertex Displacement Mapping genannt wird. Hierbei entstehen Details nicht nur als echte Geometrie auf dem Modell, sondern inklusive ihrer korrekten Verschattung. Dazu muss das 3D Objekt jedoch eine saubere Topologie sowie eine hohe Unterteilung der einzelnen Polygone aufweisen, damit die veränderte Silhouette ohne Verzerrungen generiert werden kann.⁹⁵ In Videospielen ist jedoch eher die Variante des Per-Vertex Displacement Mappings mit *Tessellation* gebräuchlich. In diesem Fall wird ein Low-Poly Mesh als Basis verwendet, welches dann im sogenannten *Tessellation Shader* als Ganzes weiter unterteilt wird. Dort wird auch die *Displacement Map* angewandt und die daraus resultierenden Vertices auf das 3D Model übertragen. Inwieweit das Basismodell jedoch unterteilt werden muss, hängt davon ab, wie weit das Objekt in der Szene vom Betrachter entfernt platziert ist. Je näher sich ein Objekt demnach am Spieler befindet, desto höher fällt der Tesselationsfaktor aus.⁹⁶

Bedient sich das Mapping jedoch der Pixel, wird von Per-Pixel Displacement Mapping gesprochen. Dieses geht eine Stufe weiter als Normal oder Bump Mapping, da es nicht nur die Flächennormalen in seine Berechnungen einbezieht, sondern zudem den Blickwinkel der Kamera hinzunimmt, um die Illusion einer komplexen volumetrischen Oberfläche zu generieren.⁹⁷ Die nötigen Berechnungen finden hierbei im *Fragment (Pixel) Shader* statt.⁹⁸ Zu diesem Verfahrenstypus gehört unter anderem als eine vereinfachte Version das Parallax Mapping, welches deswegen unter anderem auch Virtual Displacement Mapping genannt wird.⁹⁹ Anstatt wie beim Normal Mapping die Texturkoordinaten nur anhand der Normal Map zu verschieben, werden sie zusätzlich in Relation zum Betrachtungswinkel versetzt. Als Datensatz dienen dabei eine Normal Map für die Flächennormalen sowie eine Height Map für die Höhenposition des Blickwinkels.¹⁰⁰ Um herauszufinden, an welche Stelle der neue Texel gesetzt wird, wird lokalisiert, wo es zu Schnittpunkten von betrachteter Textur und der tatsächlichen Geometrie kommt. Mit Hilfe dieses Punktes und der Höheninformation der Height Map kann nun ein Vektor errechnet werden, welcher die neue Position des Texels anzeigt.¹⁰¹ Da es jedoch einfacher ist, Tiefe statt Erhebung widerzuspiegeln, wird oftmals eher die Inverse einer Height Map genutzt. Dadurch wird an der Stelle optisch Tiefe auf dem 3D Objekt erzeugt, welche der Betrachter anvisiert. Auf das gesamte Mesh angewendet, erhält das Modell mehr optische Tiefe.¹⁰²

⁹⁵ Szirmay-Kalos, Umenhoffer (2006), S. 1-6

⁹⁶ Stockmann (2017)

⁹⁷ Zink (2013)

⁹⁸ Tartachuk (2012), S. 12

⁹⁹ Polycount (2017)[A]

¹⁰⁰ Balazs (2017)

¹⁰¹ Recker (2006), S. 20-21

¹⁰² De Vires (2015), S. 392-395

Aufbauend auf die Grundprinzipien des Parallax Mappings gibt es weiterhin verschiedene Ableger dieser Methode, wie beispielsweise das Parallax Occlusion Mapping. Im Gegensatz zum Parallax Mapping tastet ein *Ray-Tracer* in umgekehrter Richtung Texture Map und Geometrie ab, um die Schnittpunkte zu finden. Dabei wird außerdem berücksichtigt, inwieweit sich die Texel untereinander aufgrund der ihnen zu Grunde liegenden Geometrie bedingen. Dieser Ansatz führt letztendlich zum Verwurf von Texeln, sollten diese in einem Bereich des Meshes liegen, welcher von anderen Teilen des Models verdeckt wird. Dadurch erscheinen beispielsweise bei einer Erhebung, wie einem herausstehenden Backstein, die der Kamera zugewandten Polygone herausragender als solche, die dem Betrachter abgewandt sind, wodurch überstehende Bereiche andere scheinbar verdecken und überschatten.¹⁰³

Ein anderer Ansatz, das Per-Pixel Displacement Mapping with Distance Function nutzt einen Algorithmus, bei welchem eine dreidimensionale Distance Map generiert wird. Anstatt nun die Schnittpunkte einer Height Map und der Geometrie des Meshes zu lokalisieren, übernimmt ein *Sphere-Tracer* diesen Schritt. Dabei wird zunächst von einem Ausgangspunkt sphärisch nach einem Schnittpunkt mit der Geometrie gesucht. Stößt die „Außenwand“ der Kugel auf einen solchen Punkt, entsteht aus dem Vektor des Blickwinkels und der Hülle der Kugel ein Berührungspunkt, welcher nun als der neue Ausgangspunkt für die nächste Iteration des Algorithmus fungiert. Dieser Schritt wird nun so oft wiederholt, bis der Schnittpunkt des Blickwinkels mit der betrachteten Position auf dem Mesh gefunden ist.¹⁰⁴

¹⁰³ Unity Technologies (2017)[D]

¹⁰⁴ Donnelly (2005), S. 126-129

	Bump Mapping	Normal Mapping	Parallax Mapping	Parallax Occlusion Mapping	Per-Vertex Displacement Mapping	Per-Pixel Displacement Mapping mit Distance-Function
Auflösung Geometrie	gering	gering	gering	gering	hoch bzw. gering (bei Einsatz von Tesselation)	gering
Änderung am Mesh	nein	nein	nein	nein	ja	nein
Datensatz	Height Map	Normal Map	Height Map, (Normal Map)	Height Map, Normal Map	Height Map (Displacement Map)	Distance Map
Dimension	2D	2D	2D	2D	2D	3D
Farbtiefe	8 bit	24 bit	8 bit, (24 bit)	8 bit, 24 bit	8 bit	24 bit
Parallaxenverschiebung	nein	nein	ja	ja	nein	ja
Shader Stage des Texturzugriffes	Pixel Shader	Pixel Shader	Pixel Shader	Pixel Shader	Vertex Shader bzw. Tesselation Shader (bei Einsatz von Tesselation)	Pixel Shader
Render Passes	1	1	1	1	1	2
Surface Rendering	down	down	down	down	up	down
Eignung für Strukturen mit hoher Amplitude	nein	nein	nein	ja	ja	ja
Selbstverschattung	nein	nein	nein	ja	ja	ja
Silhouettenbildung	nein	nein	nein	nein	ja	nein
Ray-Tracing	nein	nein	nein	Ja (Reverse Height Field Tracing)	nein	Ja (Sphere Tracing)
Verwurf von Texel	nein	nein	nein	ja	Ja (durch Z-Test, nicht Mapping)	ja

Tabelle 2: Gegenüberstellung von Surface-Detail Mapping Methoden hinsichtlich Funktion und Anwendung

4 Chancen und Limits von Texture Maps

4.1 Technischer Ansatz

Obwohl sich hauptsächlich der Designer mit Texture Maps beschäftigt, stehen nicht nur Grafik orientierte Entscheidungen hinter ihrer Verwendung. So sind auch Texturen wie alle anderen Game Assets mit gewissen *Performance*-Kosten verbunden, welche im Vorfeld und während der Entwicklung der Spielinhalte gegen den tatsächlichen Nutzen abgeschätzt werden müssen.

4.1.1 Texture Map und 3D Model

Insofern, dass Texture Maps die Oberfläche gestalten, die 3D Objekte ihnen darlegen, können diese beiden Instanzen auch hinsichtlich der Performance nicht voneinander getrennt betrachtet werden. Die Textur hat in dieser Symbiose die Aufgabe, das Detail darzustellen, wozu das Model aufgrund des beschränkten Polycounts nicht in der Lage ist. Denn je mehr Polygone in einer Szene sichtbar sind, desto höher ist der Rechen- und Speicheraufwand diese darzustellen. Texture Maps stellen also sicher, dass die Performance nicht durch einen zu hohen Polycount gedrosselt wird. Dabei muss jedoch beachtet werden, dass sie letztendlich nur zweidimensionale Bilddateien verkörpern, die wirkliche dreidimensionale Geometrie nicht ersetzen können. Daher muss der Designer das Mesh mit so vielen Polygonen ausstatten, wie nötig sind, um seine Form authentisch zu beschreiben, welche das Texturing im Anschluss noch verfeinert. Eine Säule kann beispielsweise aus einem Zylinder erzeugt werden, welcher deren abgerundeten Mittelteil bereits richtig darstellen kann. Wird nun dazu in der 3D Software Autodesk Maya 2016 der Grundkörper ausgewählt und für das Texturing mit einer zusätzlichen Unterteilung in der Mitte gemappt, besitzt dieser standardmäßig 60 Edges und damit einen Polycount von 40 Quads. Die Form der Säule wird bei dieser Ausführung sichtlich authentisch dargestellt. Reduziert der Designer diese nun um die Hälfte, verringert sich die Anzahl der Polygone schon auf 20. Trotz der geringeren Zahl ist die Form nicht verfälscht, lediglich am Übergang von Säulenrumpf zum Sockel fallen die reduzierten Edges ins Auge. Diese Variante des Objekts ist jedoch durch den geringeren Polycount viel sparsamer bezüglich der Performance. Auch wenn 20 Faces weniger nicht besonders einsparender klingen, auf die große Masse an Assets in einer Szene gerechnet, rentiert sich die Kürzung jedoch immens. Der Trick, um die geringere Unterteilung zu kaschieren, liegt in der Art, wie die einzelnen Edges geshadet sind. In der Realität besitzt kaum eine Kante eine Tiefe von null; diesen Ansatz greift das Edge Shading auf, mit dessen Hilfe eine Edge *soft* oder *hard* dargestellt werden kann. Bei letzterem erscheint sie mit einer Tiefe von null, während sie auf *soft* geshadet weicher beleuchtet wird und so wirkt, als hätte sie eine Tiefe größer als null. Durch diese Einstellung kann folglich auch eine nicht zur Gänze durch Unterteilung gebildete Rundform glatter erscheinen. Mit dieser Methode sollte allerdings vorsichtig umgegangen werden, da sie bei zu wenigen Polygonen nicht mehr zufriedenstellende Resultate erzielen kann und die Form unnatürlich verfälscht wird.

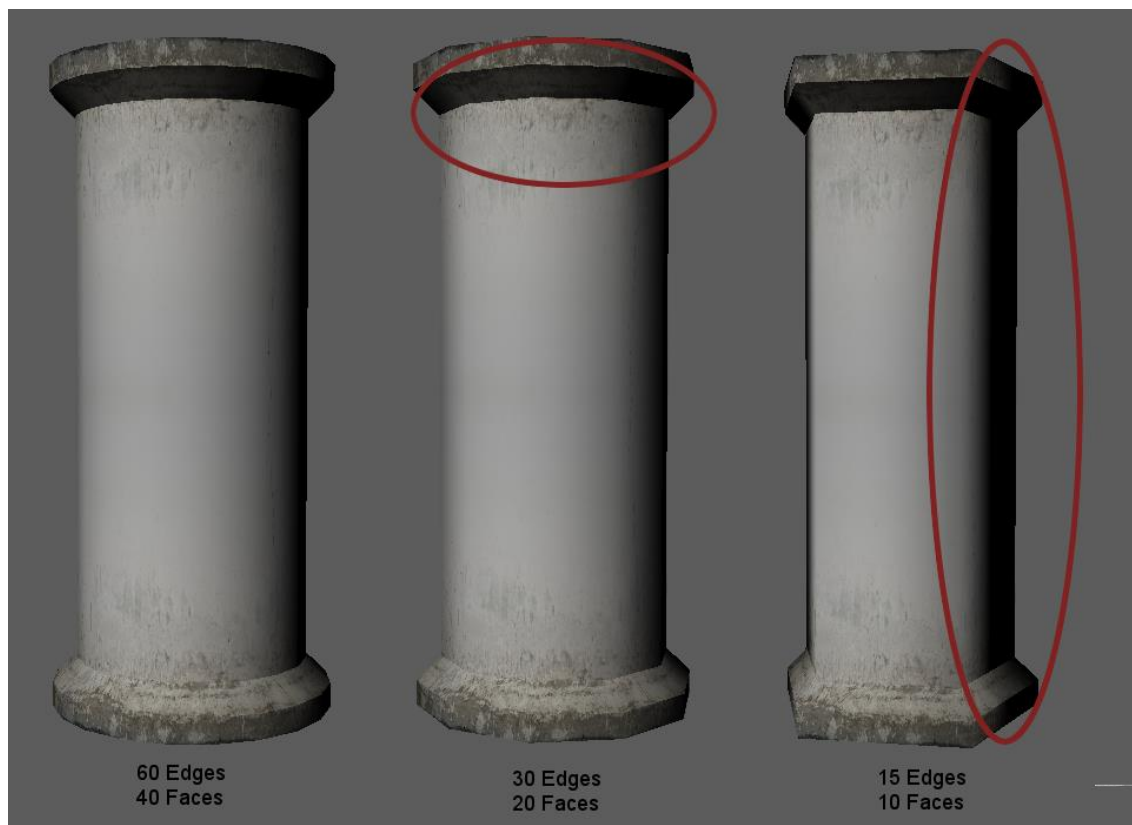


Abbildung 2: Ausführungen einer Säule mit verschiedenem Polycount

4.1.2 Priorisierung von Game Assets

Bei der Entwicklung dreidimensionaler Game Assets wird somit im Allgemeinen ein relativ, zu den Modellen abgestimmter, geringer Polycount angestrebt. Allerdings ist dies zusätzlich davon abhängig, wie Assets im Spiel verwendet werden sollen. Einige der Modelle fließen als Umgebung in die Spielwelt ein, wie beispielsweise Klippen, Wände oder Zäune. Wieder andere fungieren als Füllelemente für die Welt, darunter fallen mitunter Möbel oder andere vom Menschen gefertigte Gegenstände. Schlussendlich nimmt der Spieler am stärksten die Charaktere, Lebewesen oder einzigartige Objekte wahr. Je nachdem, in welche Kategorie das Asset fällt, wird es zum Teil anders texturiert. Zunächst einmal werden priorisierte Modelle der letzten Kategorie meist nicht nur mit mehr Polygonen gemodelt, sondern auch mit höher aufgelösten Texturen sowie generell mehreren verschiedenen Texture Maps ausgestattet. An dieser Stelle wird daher mehr investiert, da der Spieler mit diesen 3D Objekten häufiger interagiert und sie deshalb auch näher und öfter sieht. Somit erhalten beispielsweise Spielfiguren meistens den stärksten Fokus sowie auch den höchsten Aufwand und Akribie im Design. In diesen Fällen – sowie auch bei Game Assets, welche eine signifikante Zusammensetzung von Materialien aufweisen, welches in dieser Form nicht noch einmal bei einem anderen Objekt auftauchen wird – legt der Entwickler das UV Mapping als sogenanntes *Hero Asset* an. Dabei werden die Polygone des 3D Modells so gemappt, dass so viel Detail wie möglich in die Texture Map einfließen kann. Die einzelnen Tex-

ture Shells können dann innerhalb der Textur mit der nur für sie geltenden Information versehen werden. Bei der Diffuse Map für einen Charakter werden zum Beispiel für jedes Kleidungsstück und jeden sich optisch abgrenzenden Teil des Körpers wie Arme, Bein oder Gesicht einzeln aufgefaltet. Die Texture Maps, die nun für dieses Model angefertigt werden, sind demzufolge exakt auf dessen UV Mapping zugeschnitten und können nicht mehr auf ein anderes Modell projiziert werden. Da bei der Generierung von einer großen Massen an 3D Objekten jedoch so viel Speicher wie möglich gespart werden will, werden Elemente, die ein und dasselbe Material nutzen, als *Standard Assets* angelegt. Hierbei mappt der Designer das Objekt so, dass so wenig Seams wie möglich entstehen, da dem Mesh anschließend eine *tileable Texture* übergeben wird. Diese Art von Textur ist so aufgebaut, dass, wenn ihre Ränder aneinander gelegt werden, kein sichtbarer Seam entsteht. So kann eine 1024x1024 Pixel Bilddatei eine 100 m² große Rasenfläche abdecken, ohne dass störende Kanten sichtbar sind. Diese Art von Game Asset wird vor allem bei Umgebungselementen oder Möbeln verwendet, da hierbei mehrere Objekte dasselbe Material verwenden. Auf diese Art kann Speicherbedarf gespart werden, da weniger Texture Maps eingelesen und gerendert werden müssen.

4.1.3 Kosten-Nutzen-Entscheidungen bei Surface-Detail Maps

Als besonders attraktiv stellen sich für den Designer die Texture Maps dar, mit denen Detail auf der Oberfläche des Models dargestellt werden kann. Aufgrund der notwendigen Priorisierung von Game Assets ist allerdings ein sparsames Umgehen mit solchen Texturen ratsam. Neben den langwierig bekannten Surface-Detail Mapping Methoden haben sich mitunter verschiedene Hardware-Hersteller wie *Nvidia* oder *ATI* der Entwicklung neuer, verbesserter Verfahren gewidmet. Jeder der existierenden Typen weist dabei gewissen Vorteile und Nachteile auf.

Die grundlegende Idee hinter all diesen Maps leitet sich zunächst aus der Existenz eines Mapping Verfahrens ab, welches zwar auch in diese Typen-Familie gehört, sich jedoch vor allem in einem signifikantem Merkmal von allen anderen unterscheidet – das Per-Vertex Displacement Mapping. Hierbei wird tatsächlich die Geometrie des 3D Models verändert, sodass eine sich selbst-verschattende, volumetrische Silhouette entsteht. Das Resultat – zumal der verwendete Datensatz nur eine Height Map beinhaltet – kommt dem eines händisch gemodelten Meshes sehr nah und liefert optisch hervorragende Resultate. Besonders bei Strukturen mit hoher Amplitude, wie beispielsweise Text oder einer Backsteinmauer mit tiefen Lücken zwischen den einzelnen Steinen, bilden sich keine sichtbaren *Artefakte* aus verschiedenen Blickwinkeln.¹⁰⁵ Al-

¹⁰⁵ Donnelly (2005), S. 123-124

lerdings ist zur Durchführung dieser Methode eine hohe Unterteilung der Polygone notwendig, damit die in der Height Map gespeicherten Höheninformationen sauber auf die Vertices des Meshes übertragen werden können. Angewendet auf mehrere Assets würde dieser Ansatz zu viel Rechenleistung verbrauchen, als dass er umfassend für eine Echtzeit-Anwendung wie ein Videospiel verwendet werden könnte.¹⁰⁶ Diese Problematik nehmen sich letztendlich alle anderen *Surface-Detail* Methoden an, weswegen sie die Geometrie des Models nicht manipulieren, sondern Detail simulieren.

Eine Möglichkeit zu entscheiden, welches Texturing-Verfahren für die Oberflächengestaltung angewendet werden soll, stellt die Frage nach dem Datensatz dar. Da dieser durch die Texture Map repräsentiert wird, ist dies vor allem eine Frage des Speicheraufwandes, da alle vom Spieler sichtbaren Texturen als Bilddateien in den Grafikspeicher geladen werden müssen. Die erste Entscheidung geht daher mit der Anzahl der verwendeten Texture Maps einher; je mehr Maps für ein Verfahren benötigt werden, desto mehr Speicheraufwand muss betrieben werden, was auf eine große Menge an Assets gerechnet die Performance schmälert. Ein Mapping Typus, der demnach nur eine Bilddatei nutzt, stellt demnach erst einmal eine speichergünstigere Variante dar. In diese Kategorie fallen somit zunächst das Bump und das Normal Mapping. Bei beiden Verfahren werden lediglich die Informationen über die Flächennormalen in einer zweidimensionalen Datei zur Darstellung des Details verwendet. An dieser Stelle kommt der erste grundlegende Unterschied der beiden Methoden zu tragen; das Bump Mapping verwendet eine Graustufen Bilddatei, wobei hingegen das Normal Mapping Farben des RGB-Farbmodells verwendet. In letzterem werden demzufolge in einem Pixel drei Farben a 256 Werte gespeichert. Eine Height Map nutzt im Gegensatz dazu keine Farbe, sondern nur den Helligkeitswert des HSB-Modells, weswegen hierbei lediglich 256 Werte pro Pixel zum Einsatz kommen. Speichertechnisch gesehen stellt demnach die Height Map die günstigere Variante dar. Andererseits ist der zweite große Unterschied zwischen beiden Verfahren nicht zu vernachlässigen; während die Normal Map bereits die Flächennormalen als vollwertige Vektoren beinhaltet, welche nur noch die bisherigen Normalen des Meshes ersetzen müssen, müssen sie beim Bump Mapping erst noch berechnet werden, da die Height Map nur die veränderte Höhenposition widerspiegelt. Bei dieser Problematik muss die Wahl der Texture Map folglich anhand von Speicherkapazität oder Rechenleistung getroffen werden. Beide Verfahren stellen nichtsdestotrotz die günstigste Alternative zu anderen Mapping Methoden dar, wobei die geringen Performance-Kosten gewisse Nachteile mit sich bringen. Dadurch dass lediglich die Flächennormalen in die Berechnung einfließen, erschienen die simulierten Erhebungen oder Dellen aus bestimmten, vor allem steilen, Blickwinkeln verzerrt. Auch kann deshalb keine volumetrische Silhouette entstehen, weshalb Normal und Bump Mapping eher für geringe Materialstrukturen wie Haut, Leder, Putz oder Ähnliches geeignet sind.

¹⁰⁶ De Vries (2015), S. 393

Eine weiter ausgereifte Methode stellt hierbei das Parallax Mapping dar. Dadurch, dass es neben den Normalen ebenfalls den Blickwinkel des Betrachters einbezieht, können authentischere Resultate erzielt werden als bei Normal oder Bump Mapping. Hierbei fällt der Datensatz aufgrund der zusätzlichen Informationen etwas höher aus. Entweder werden eine Height Map für die Flächennormalen und eine Normal Map für die Parallaxe verwendet oder aber der Vektor des Blickwinkels wird innerhalb der Height Map gespeichert. Bei letzterem wird zwar nur eine Map verwendet und somit auch weniger Speicher, allerdings muss im Gegensatz zu der ersten Variante der Betrachtungswinkel nachträglich noch aus den Werten der Height Map berechnet werden. Auch hier begegnet der Designer der allzeit präsenten Frage nach Rechenleistung oder Speicheraufwand. In Hinblick auf die beiden simpleren Methoden müssen beim Parallax Mapping sogar verhältnismäßig geringfügig mehr Schritte im Shader durchlaufen werden, sodass an dieser Stelle der Aufwand nicht großartig ansteigt.¹⁰⁷ Auch wenn diese Methode ähnlich schnell dem Normal und Bump Mapping arbeitet und einen besseren Eindruck von Tiefe verschaffen kann, erreicht dieses Mapping ebenfalls seine Grenzen bei hochfrequenter Geometrie. Stumpfe Betrachtungswinkel bereiten dabei keine Probleme, bei steilen Winkeln bilden sich jedoch auch hier verzerrte Kanten. Diese bilden sich deshalb, da das Parallax Mapping nur grob Schnittpunkte von Textur und Mesh lokalisieren kann und dadurch Lücken in dem gerenderten Model auftreten. Dieses Phänomen wird als *Oversampling* bezeichnet und wird seither durch verbesserte Modelle versucht zu umgehen.

Einer dieser Versuche endete im Aufkommen des Parallax Occlusion Mappings, welches den gleichen Datensatz nutzt wie das Parallax Mapping, jedoch durch den Einsatz des Ray-Tracers viel mehr Rechenaufwand betrieben werden muss. Die Ergebnisse fallen dementsprechend qualitativ aus; durch das genauere Abtasten können exaktere Texturkoordinaten kalkuliert werden, welche nicht nur optisch mehr Tiefe bewirken, sondern auch zum Verwurf von Texeln führt, welche ansonsten unsaubere Kanten hervorrufen würden. Ein weiterer Vorteil ergibt sich daraus vor allem in der Selbstverschattung, welche eine logische und weichere Schattenbildung ermöglicht. Jedoch kommt es auch bei Betrachtung des 3D Objects aus größerer Entfernung zu Artefakten, welche nur durch eine Erhöhung der Abtaste des Ray-Tracers gelöst werden könnte, was besonders für Videospiele in hohen Performance-Einbußen endet. Ähnlich verhält es sich mit der Silhouettenbildung, die auch in diesem Verfahren nicht zu Stande kommt und durch zusätzliche Schritte im Mapping Algorithmus generiert werden müsste.¹⁰⁸ Neben einer größeren Zahl an Rechenleistung müssen vor allem die verwendeten Normal Maps bei komplexeren Objekten sehr hoch aufgelöst sein, um eine gute Qualität beim Rendern zu erhalten.¹⁰⁹

¹⁰⁷ Tartachuk (2005), S. 18

¹⁰⁸ Ebd., S. 33-34

¹⁰⁹ Ebd., S. 62

Ähnliche Resultate erzeugt ebenfalls das Per-Pixel Displacement Mapping with Distance Function. Der wohl augenscheinlichste Unterschied stellt dabei die Verwendung einer dreidimensionalen Distance Map dar, welche im Vergleich zum Datensatz der anderen Methoden drei Informationen des 3D Raums abbildet; so wird in diesem Fall auch die Tiefe benötigt, statt nur Länge mal Breite, was einen höheren Speicheraufwand mit sich zieht. Allerdings ist das Sphere-Tracing ein schnelles Verfahren, was auf der Seite der Rechenzeit dem Reverse Ray Tracing des Parallax Occlusion Mappings wieder entgegenwirkt.

Rückwirkend stellt sich nun die Frage, warum das Per-Vertex Displacement Mapping trotzdem als anerkanntes Verfahren gebraucht wird. Die Antwort darauf hängt zum einen mit der Verteilung der Assets in der Szene zusammen, zum anderen mit der Funktionsweise der Methode. Zunächst verwendet dieses Displacement Mapping lediglich eine 8 bit Graustufen Textur, welche zwar eine relativ hohe Auflösung für einen entsprechend hohen Detailgrad aufweisen muss, jedoch immer noch speicherschonender ist, als die 3D Distance Map des Per-Pixel Displacement Mapping with Distance Functions oder die Map-Kombinationen aus Normal und Height Map beim Parallax Mapping und Parallax Occlusion Mapping. Weiterhin finden die Operationen, die das Mesh unterteilen und manipulieren im Tessellation Shader statt. Einerseits werden auf diese Weise die Shading Einheiten nicht belastet und andererseits beschränken sich die Rechenschritte auf einfache Vertextransformationen wie Rotation oder Verschiebung. Diese sind weitaus sparsamer in Bezug auf die Performance als aufwendige Ray-Tracing Verfahren. Der große Nachteil dieses Algorithmus liegt jedoch in der Bestimmung der notwendigen Unterteilungen des 3D Models, welche von der Entfernung des Objekts zum Betrachter abhängig ist. Als problematisch stellt sich dieser Umstand heraus, wenn es darum geht, Objekte, die sich über Vorder- und Hintergrund der Szene erstrecken, mit dieser Methode zu rendern. Der Teil des Meshes, der sich dabei in unmittelbarer Umgebung des Spielers befindet, würde einen hohen Tessellationsfaktor benötigen, um einen hohen Detailgrad darstellen zu können, während der Bereich im Hintergrund aus dem direkten Sichtfeld des Betrachters schwindet und weniger Detail benötigt. Da die Tessellation auf das 3D Objekt in seiner Gesamtheit angewendet wird, würden hier unnötig viele Polygone erzeugt werden. Eine weitere kontraproduktive Auswirkung findet sich dann, wenn ein Modell mehrfach in der Szene auftaucht und sich dabei in unterschiedlichem Abstand zum Spieler befindet, zum Beispiel Fässer oder Kisten. In diesem Fall müsste für jedes Objekt der Tessellationsfaktor neu bestimmt werden und diese Abfragen kosten entsprechend Rechenzeit. Vergleichsweise dazu sind Normal Mapping und Co für solche Art Game Assets besser geeignet, da hierbei der Detailgrad im Fragment Shader bestimmt wird; je weiter der Teil eines Models vom Betrachter entfernt ist, desto geringer ist dessen Auflösung in Pixeln und somit insgesamt auch die Performance.

Resultierend lässt sich demnach sagen, dass Geometrie manipulierende Verfahren wie das Per-Vertex Displacement Mapping aufgrund ihrer hoch-qualitativen Resultate und Tessellation eher für detaillierte, nur einmal auftauchende Game Assets wie beispielsweise Charaktere angewendet werden sollte. Ansätze, die jedoch im Pixel Shader agieren und Oberflächendetail simulieren, eignen sich durch die automatische Redu-

zierung der Auflösung mit zunehmender Entfernung und einem geringeren Polycount für Game Assets der Umgebung, Sammelobjekte oder allgemein mehrfach genutzte 3D Modelle.

4.2 Narrativer Ansatz

Insofern dass Texturen nicht nur aus unsichtbarem *Code* bestehen, sondern sich vor allem durch ihr visuelles Feedback an den Betrachter auszeichnen, ist der Designer in der Lage, durch sie gezielte Emotionen und Reize beim Spieler zu erzeugen. Texture Maps sind demnach neben anderen Bestandteilen eines Videospiels, wie Audio, Gameplay oder Animationen, ein essenzielles Werkzeug, um eine Geschichte zu erzählen. Visuelle Narration zieht sich dabei durch das gesamte Spiel mit all seinen Game Assets und verdrängt den Eindruck, das Geschehen nur von außen zu betrachten. Durch das sich ständig wechselnde Design der Umgebung begleitet den Spieler eine andauernde Entwicklung der Spielwelt, die nicht nur ihre Vergangenheit, sondern auch die Gegenwart, in der sich der Spieler bewegt, dokumentiert.¹¹⁰

4.2.1 Narration durch Farbe

Damit das Vermitteln der Geschichte gelingt, ist in den meisten Fällen dreidimensionaler Videospiele die Kombination mehrerer Texture Maps nötig. Primär erfolgt die Bewertung einer Spielwelt vor allem über die Diffuse Map, da sie die Farbe eines Objektes repräsentiert, welche vom Menschen zunächst am bewusstesten wahrgenommen wird. Farbe ist somit eine bedeutsame Schnittstelle zwischen Betrachter und Spiel, wobei sie jedoch auch am schwersten zu kontrollieren ist. Jeder Mensch besitzt aufgrund verschiedenster Faktoren eine von anderen Menschen abweichende Farbauffassung. Hierbei können die persönliche Vorgeschichte, der kulturelle Hintergrund oder Unterschiede im Verarbeitungsprozess von Farbe durch das menschliche Auge Grund für diesen Effekt sein.¹¹¹ Manchen Farben kann allerdings auch ein symbolischer Charakter zugeordnet werden, welcher das Erkennen der Aussage der entsprechenden Farbe weitestgehend universell werden lässt.¹¹²

So haben sich bestimmte Farben auch in der Videospielebranche etabliert. Besonders die Farben rot, gelb und grün können im westlichen Kulturkreis eindeutige Emotionen zugeschrieben werden. Wie auch bei einer Ampel repräsentiert rot Gefahr, gelb für Vorsicht und grün für das Fehlen von Gefahr. Am Beispiel des Gegnertypus „Big Dad-

¹¹⁰ Ebd.

¹¹¹ Hammer (2008), S. 156-157, 183

¹¹² Bartel (2013), S. 42

dy“ im *Shooter BioShock* weist ebenfalls je die Färbung des Kopfteils auf den Zustand des Charakters hin. Leuchtet es rot, befindet sich der Gegner im Angriffsmodus, so dass der Spieler von einer Gefahr für seine Spielfigur ausgehen muss. Gelb symbolisiert weiterhin eine neutrale Verfassung, bei dem der Widersacher den Spieler noch nicht als Bedrohung wahrnimmt und ihn nicht angreifen wird. Letztendlich zeigt grün die freundliche Gesinnung der Figur an, obgleich im Fall des „Big Daddy“ dieser sogar zum Verbündeten wird.¹¹³



Abbildung 3: Zustände des "Big Daddy" in *BioShock*

Die Verwendung der gleichen Farbgebungen für bestimmte Elemente macht es dem Designer einfacher, dem Spieler determinierte Empfindungen zu vermitteln und somit eine abschätzbare Reaktion zu erzeugen. Auf diese Weise beobachtet er das Geschehen nicht nur, sondern nimmt aktiv daran teil. Diese Herangehensweise im Umgang mit Farbe erfordert jedoch nicht nur vom Entwickler selbst eine Gewissenhaftigkeit, diese verlangt das Spiel im Gegenzug ebenfalls vom Spieler. Denn Farbe kann zwar auf ein und dieselbe Weise verstanden werden, allerdings ändert sich ihre Bedeutung im Zuge verschiedener Kontexte. Je nachdem, welcher Verständnisbereich des Spielers angesprochen werden soll, ändert sich auch die Wahrnehmung von Farbe. Im Bereich Gegner wird im Fall des „Big Daddy“ rot als Bedrohung empfunden, während die gleiche

¹¹³ Fandom (2017)[B]

Farbe im Kontext der eigenen Spielfigur als Lebenskraft aufgefasst wird. Letzteres erfährt der Spieler in Form des sogenannten „ADAM“, eine Flüssigkeit, welche der Spieler sammeln und in den Ausbau der Fähigkeiten seines Charakters investieren kann.¹¹⁴



Abbildung 4: ADAM Injektor in BioShock

Besonders in Hinsicht auf Game Assets, auf welche der Spieler reagieren muss, um im Geschehen voranzuschreiten, weisen die Farben in der Diffuse Map oft einen hohen Eigenwert auf. Die Farbe referiert hierbei auf sich selbst und ihren Symbolcharakter, anstatt auf die tatsächliche Farbe des Objekts.¹¹⁵ Trotz dass eine medizinische Flüssigkeit in der Realität niemals rot wäre, erkennt der Spieler im Fall von *BioShock* das „ADAM“ als solche an, da er gelernt hat, dass Rot in diesem Zusammenhang eine positive Auswirkung für ihn zur Folge hat. Dieser Effekt lässt sich sogar auf ganze Levelbereiche beziehen. Im Gegensatz dazu kann Farbe außerdem genutzt werden, um ein Objekt in seiner natürlichen Färbung wiederzugeben. An dieser Stelle spricht der Designer vom Darstellungswert der Farbe, welche infolgedessen dem entsprechenden Gegenstand komplett untergeordnet ist.¹¹⁶ Dem Designer obliegt es nun, die Balance zwischen Diffuse Maps mit einem hohen Grad an Eigen- oder Darstellungswert zu finden. Der Eigenwert sollte dabei eher sparsam verwendet und größtenteils auf die grundlegenden Spielmechaniken angewendet werden, um den Spieler nicht mit Reizen zu überfluten und sein Verständnis der Elemente im Spiel nicht zu stören. Denn obwohl der Darstellungswert nur die Objekte auf die Weise widerspiegelt, wie sie wirklich be-

¹¹⁴ Fandom (2017)[A]

¹¹⁵ Gage (1999), S. 50

¹¹⁶ Ebd.

schaffen sind, ist das genau seine Stärke. Trotz der herausstechenden und spielentscheidenden Assets muss die Spielwelt in einem gewissen Grad in der Realität verankert sein und dem Spieler authentische und vertraute Visualisierungen liefern. Denn nur, wenn es etwas Bekanntes gibt, kann das Ungewöhnliche auch wahrgenommen werden. Dieses Konzept aus farbllichem Charakterdualismus kann letztendlich sogar auf das gesamte Spiel bezogen werden. Wird das Niveau an Darstellungswert über die komplette Spielwelt eingehalten, können ortsspezifische Farbschemata ein mentales Bild beim Spieler erzeugen.¹¹⁷ In *Batman: Arkham City* zeigt die Spielwelt durchgehend ein einheitlich architektonisches Stadtbild, obgleich die verschiedenen Einrichtungen, in denen sich zentrale Ausläufe der Handlung im Spiel zutragen, ein prägnantes Farbschema aufzeigen. Der Spieler erkennt hierbei beispielsweise das Stahlwerk an seinen schmutzigen Braun- und Orangetönen sowie die Iceberg Lounge anhand der kühlen Blaunuanzen.

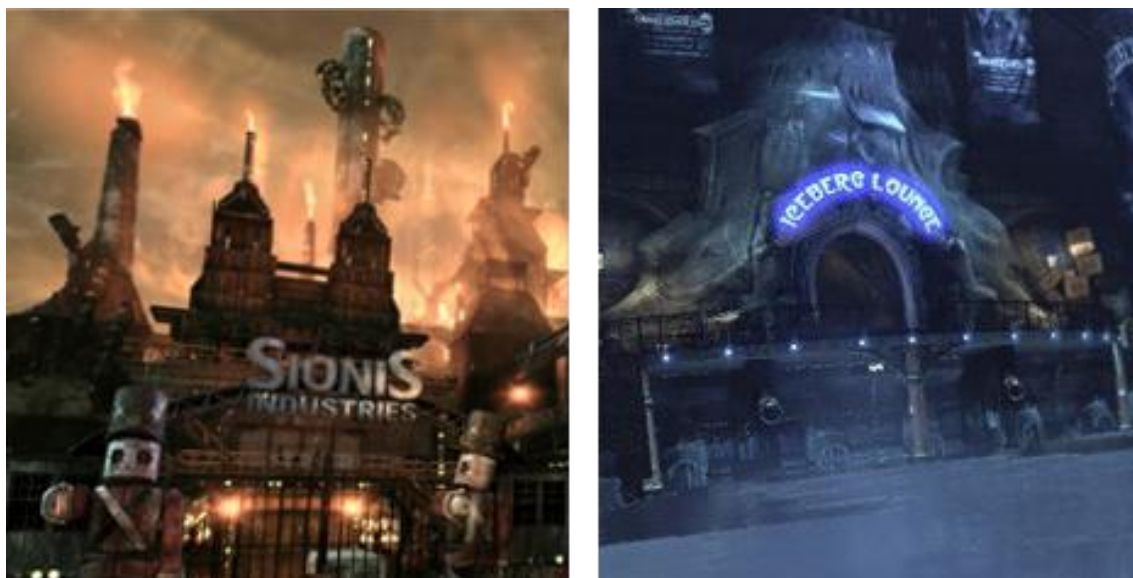


Abbildung 5: Farbschemata durch Texture Maps in *Batman: Arkham City*

Ein weiterer Aspekt in der menschlichen Farbwahrnehmung, die sich der Designer zu Nutze machen kann, ist das Empfinden von warmen und kalten Farben. Ob nun als Eigen- oder Darstellungswert erscheinen wie auch für den Betrachter eines Gemäldes dem Spieler Rot und Gelb als warme Töne, zu denen er sich hingezogen und sicher fühlt, während Blau und Grün zur kühlen Farbpalette zählen und oft als düster oder mysteriös aufgefasst werden. In zahlreichen Videospielen wird beispielsweise ein Feuer, untermalt durch seine warmen Rot- und Orangetöne, zur Zufluchtsstätte, an der der

¹¹⁷ Wallace (2015), S. 139

Spieler keine Gefahren zu befürchten hat. Oft dienen sie zudem als Checkpoint, bei welchem das Spiel gespeichert wird.



Abbildung 6: Lagerfeuer in *Horizon Zero Dawn*, *Dark Souls* und *Tomb Raider*

Verstärkt werden kann dieses Phänomen weiterhin durch die Verwendung von Simultankontrasten. Eine Farbe kann bei Präsenz ihrer Komplementärfarbe heller, dunkler oder intensiver hervortreten, so wie weiß heller erscheint, wenn es von schwarz begrenzt ist und andersherum.¹¹⁸ Spiegelt sich dieses Prinzip an geeigneten Game Assets wieder, kann auch an dieser Stelle kontrolliert Empfindungen beim Spieler ausgelöst werden. In prekären Situationen können diese Reizerzeuger wichtige Hinweise liefern, die in der detaillierten Umgebung ansonsten möglicherweise untergegangen wären.¹¹⁹ Auf einer grünen Wiese können beispielsweise rote Blumen, die oft Heilkräuter symbolisieren, schneller entdeckt werden, was dem Spieler das Überleben seiner Spielfigur ermöglichen kann, falls dessen Gesundheit sehr niedrig ist. Interessante Orte oder Gegenstände können auf diese Weise in einer durch Schnee und Berge bläulich erscheinende Umgebung die Aufmerksamkeit des Spielers auf sich ziehen, wenn sie orange gefärbt sind. Oft sind es jedoch sogar die elementarsten Manipulationsmöglichkeiten, die den größten Effekt erzielen können. Allein, indem Farbton, Helligkeit und Sättigung einer Farbe verändert werden, kann sich der Charakter eines Objektes grundlegend ändern. In der Erweiterung *Tiny Tina's Assault on Dragon Keep* des Shooters *Borderlands 2* ändert die Spielfigur Tiny Tina eine ganze Umgebung in ihrer Erzählung von freundlich und märchenhaft zu düster und gruselig. Optisch werden neben ein paar Modellen und das Licht zum Großteil die Farben justiert, um dieser Geschichte gerecht zu werden. Die Brauntöne verfärben sich ins Bläuliche und Sättigung sowie Helligkeit werden herabgesetzt.

¹¹⁸ Bonanomi, Rizzi (2017), S. 35

¹¹⁹ Wallace (2015), S. 71



Abbildung 7: Unverändertes und manipuliertes Startareal in Borderlands 2: Tiny Tina's Assault on Dragon Keep

Das Fehlen von Sättigung und Helligkeit muss allerdings nicht zwangsläufig negativ ankommen, es kann in manchen Zusammensetzungen auch edel und geheimnisumwoben wirken. So erscheinen teilweise auch übersättigte Farben dem Spieler als surreal, überreizend oder quälend. Es kommt in jedweden Fall auf die Kombination aus diesen drei Faktoren an, die den Charakter einer Farbe ausmacht.

4.2.2 Narration durch Licht und Schatten

Trotz der menschlichen Fähigkeit Farben in unterschiedlichen Situationen konstant wahrnehmen zu können, beeinflusst eine weitere Größe die Auffassung von Objekten – Licht. Der Mensch kann Licht allerdings erst erkennen, wenn ein Körper auf es reagiert. Diese Reaktion erfolgt in zwei Schritten; einerseits im Ausbilden von Schatten am Gegenstand sowie um ihn herum und andererseits als direkte Reaktion des Lichts mit der Oberfläche. Bei letzterem kann Licht entweder gestreut, reflektiert, absorbiert oder durchgelassen werden, je nachdem, wie das Material des Objektes beschaffen ist.¹²⁰ Der Designer kann sich diesem Phänomens bemächtigen, indem er mittels der Specular und der Occlusion Map den Charakter des Materials eines Modells definiert.

Je nachdem, wie groß der Fokus des Spielers auf ein Asset sein soll - ob es sich entweder um ein Aufmerksamkeit erregendes Objekt oder eher ein Füllelement in der Umgebung handelt – werden Lichteffekte unterschiedlich stark umgesetzt. Interagierbare oder zentrale Modelle erfahren dabei aufwendigere Effekte per benutzerdefinierte Shader als Gegenstände, welche nur der Dekoration dienen.¹²¹ Doch selbst für letztere bieten zunächst die Specular Map sowie die Roughness Map großes Potenzial, die Geschichte hinter dem Objekt zu erzählen. Ein Holzfußboden in einem Haus kann bei-

¹²⁰ LEIFphysik (2017)

¹²¹ Ahearn (2006), S. 32

spielsweise schon seit langem in Benutzung sein; viele Menschen sind möglicherweise bereits über ihn gelaufen und haben das Holz dadurch an manchen Stellen mehr abgelaufen als an anderen, weswegen er an diesen besonderen Stellen glänzender ist. Vielleicht haben Kinder an einer ganz bestimmten Stelle im Raum gesessen und gespielt, sodass der restliche Boden stumpfer das Licht reflektiert als in diesem Bereich. Selbst wenn die Holzpaneele gerade erst neu verlegt worden sind, gibt es Furchen in der Maserung, Astlöcher oder Schnittkanten, die anders auf Lichteinfall reagieren und dem Betrachter somit seinen Charakter verraten. Allein die Art und die Verarbeitung des verbauten Holzes und der Aufwand an Pflege durch die Bewohner kann durch die Lichtreflektionen erkannt werden. Ausgehend von diesen kleinen Brocken visueller Information ist der Spieler in der Lage sich vorzustellen, wer wohl in diesem Haus leben mag und wie der Raum mit dem Holzboden genutzt wird. Ist der Bewohner reich oder eher weniger vermögend? Leben viele oder wenige Personen an diesem Ort und ist dieser Raum für sie von besonderem Wert? Unter Rücksicht auf diese Gesichtspunkte verweist auch der Holzboden in *Rise of The Tomb Raider* auf den altherwürdigen Charakter des Croft Anwesens.



Abbildung 8: Holzboden im Croft Anwesen in Rise of the Tomb Raider

Ähnliche Erkenntnisse lassen sich auch durch den Effekt der Umgebungsverdeckung des Occlusion Mappings ableiten. Indem sich beispielsweise ungenaue Schnittkanten der Holzpaneele des Bodens gegenseitig um ein paar Millimeter verdecken, gewinnt der Spieler immer mehr den Eindruck, dass es sich hier um weniger qualitatives Holz handelt und der oder die Bewohner des Hauses wahrscheinlich weniger vermögend sein könnten. Je nach der restlichen Einrichtung kann dieser Hinweis aber auch andeuten, dass der Boden per Hand verlegt wurde und somit vielleicht ein handwerklich begabter Mensch in diesen Räumen lebt. Auch bei der Verwendung von Licht und Schatten kann man diese nie ohne einen individuellen Kontext interpretieren. Durch die

verstärkte Schattenausbildung erhalten die Modelle mehr Tiefe und somit mehr Glaubwürdigkeit, da sich ihre Oberfläche in Relation zu den einzelnen Objektbestandteilen glaubwürdig verhält. Auf diese Weise kann zusätzlich ein höherer Grad an Realismus erreicht werden.

4.2.3 Narration durch Oberflächendetail

Neben Farbe, Licht und Schatten stellt die Möglichkeit mittels Bump, Normal oder Displacement Mapping scheinbare Oberflächendetails zu generieren die größte Stärke von Texture Maps dar, einem Objekt Individualität einzuhauchen. Selbst eine fotorealistische Diffuse Textur kann nicht über die Tatsache hinwegtäuschen, dass Game Assets flache Polygone zu Grunde liegen. Die Realität zeichnet sich jedoch durch Unebenheiten und Tiefe aus. Je mehr Feinheiten also auf dem Objekt zu sehen sind, desto mehr Informationen erhält der Spieler über die Umgebung, einen Gegenstand oder einen Charakter. Besonders in Hinsicht auf letzteren bekommen Tiefendetails monumentales Gewicht. Neben der Umgebung stechen besonders die Figuren hervor, denn es ist deren Geschichte, die der Spieler erfahren möchte. Feine, aber charaktergebende Gesichts- und Körpermerkmale lassen den Spieler nachvollziehen, wie die Geschehnisse im Spiel die Figur verändern. Durch den Fortschritt der Hardware rückt das Aufzeigen von glaubwürdiger Menschlichkeit zunehmend in den Bereich des Möglichen. An der Spielereihe *Uncharted* lässt sich dieser Prozess eindrücklich nachvollziehen. Während der Protagonist Nathan Drake im ersten Teil der Reihe durch nur wenig Falten, eine glatte Haut und eine ebenmäßige Augenpartie jung und unverbraucht wirkt, evolvieren diese Merkmale zunehmend über die Folgetitel hinweg. Es kommen Falten sowie unterschiedliche Hautschattierungen hinzu, Kopfform und Gesichtskonturen prägen sich stärker aus und erscheinen definierter. Nicht nur, dass die Figur menschlicher wird, sie altert außerdem und lässt dadurch den Rückschluss auf die bisher erlebten Abenteuer und Strapazen zu.¹²²

¹²² Meyer et. al. (2016)



Abbildung 9: Evolution von Nathan Drake in Uncharted 1-4

Doch auch für weniger zentrale 3D Modelle kann dieses Prinzip narrative Kraft entfalten. Falten in der Kleidung einer Figur können dabei nicht nur Aufschluss über die Beschaffenheit des Materials – beispielsweise bildet Seide im Vergleich zu Jeans mehr und tiefere Falten aus – geben, sondern auch über den Körper der Person, die das Kleidungsstück trägt. In diesem Fall wäre der Spieler dadurch in der Lage Gewicht und Muskulatur ablesen zu können. Und besonders dann, wenn ihm bis dahin keinerlei Informationen über diese Merkmale der Person vermittelt worden sind, beginnt der Betrachter Fragen zu stellen. Über den Hintergrund, warum eine Person zum Beispiel ausgemergelt und dürr wirkt; welche Umstände brachten sie zu so einer drastischen Veränderung? Gibt es psychische Probleme? Oder befindet sich der Mensch in einer Umgebung, die nur eine geringfügige Nahrungsaufnahme zulässt? Abgesehen von solchen eher offensichtlicheren Erscheinungen kann Oberflächentiefe auch unterstützend zu anderen Visualisierungsmöglichkeiten eingesetzt werden. In *Uncharted 4: A Thief's End* untermalen die Falten auf der Kleidung das an ihnen sichtliche tropische Klima. Sie sind deutlich kompakter, ohne weiche Ausläufe und schmaler und wirken so, als würden sie am Körper kleben.



Abbildung 10: Falten in der Kleidung als Zeugnis tropischen Klimas in Uncharted 4: A Thief's End

Diese Art von Detail erklärt sich für den Spieler von allein, da er sich parallel zur Figur in den Geschehnissen befindet. Er nimmt ebenso wie der Charakter die Umgebung wahr und fordert instinktiv die Auswirkungen optisch ebenfalls aufgezeigt zu bekommen. Ist dies der Fall, wirkt das Gesehene als authentisch empfunden und nur, wenn der Spieler das Spiel auf diese Weise verifiziert, erlaubt er auch das zuvor beschriebene Fragen nach versteckten visuellen Informationen. Und genau dieses Hinterfragen erzeugt eine unterbewusste, aber essenzielle Immersion, durch die der Spieler das Spiel erlebt.

4.3 Realismus und Stilismus

Durch die zunehmenden Möglichkeiten in der Videospielentwicklung aufgrund von höherer Rechenleistung und leistungsstärkerer Grafikhardware tendiert der Anspruch vieler Studios in Richtung Realismus.¹²³ Allerdings beinhaltet diese Begrifflichkeit ein größeres Interpretationsspektrum als zunächst angenommen wird. Der erste Gedanke gilt dabei meistens der Grafikqualität; je detailreicher und farblich korrekt, desto realistischer scheint das Spiel zu sein.¹²⁴ Realismus kann jedoch auf vier verschiedene Arten erreicht werden, welche zwar alle auf die Qualität des Visuellen referieren, jedoch nicht nur auf Farbe und Detail hinweisen. Die Realität in Videospielen setzt sich hauptsächlich aus den Proportionen (Modelle), der Oberflächenbeschaffung (Texturen), der Beleuchtung (Shader) sowie der Bewegung (Animationen) zusammen. Der Bewertungsfaktor eines jeden stellt dabei die physikalische Korrektheit dar. Bewegen

¹²³ Low (2001)

¹²⁴ Ebd.

sich Gegenstände und Charaktere so wie sie es von Natur aus tun? Stimmen die Proportionen der 3D Modelle mit ihren Vorbildern aus dem wahren Leben überein? Und reagiert die Umgebung auf die Art und Weise auf Licht, wie es normalerweise tun würde? Jeder dieser Bereiche kann einen unterschiedlich hohen Grad an Realismus aufweisen, wobei erst die Kombination aller den eigentlichen Stil des Spiels ausmacht.

4.3.1 Realismus durch Texturen

Bei dem Versuch die Realität abzubilden, ist vor allem das Zusammenspiel der verschiedenen Texture Maps unabdingbar. Dabei kommt es nicht nur auf die Qualität jeder einzelnen an, sondern ebenfalls auf die Kongruenz der Typen untereinander. Eine Specular Map, die nicht konkret für die vorangegangene Diffuse Map angelegt wurde, verfälscht die Wirkung im Endergebnis, da Teile des Mesh fälschlicherweise stärker auf Licht reagieren, als ihre Natur es eigentlich hergibt. Um solche Missstände zu vermeiden, werden die meisten Texturen direkt aus einem 3D Modelling Programm extrahiert, da Licht und Schatten direkt in der Applikation vorkalkuliert werden und somit exakter auf die Diffuse Map passen, als wenn sie händisch erzeugt werden würden. Das volle Potenzial von Texture Maps kann allerdings erst durch den Shader freigesetzt werden, da eine Materialoberfläche erst durch natürlich Ausleuchtung wirken kann. Genau auf diesen Aspekt konzentriert sich das *Physically-based Rendering (PBR)*, welches seit geraumer Zeit großen Anklang in der Entwicklung von Videospielen mit realistischer Ambition findet. Um Licht und seine Auswirkung auf eine Szene physikalisch so korrekt wie möglich darstellen zu können, verwendet der PBR Ansatz in sich mehrere Rendering-Methoden. Ähnlich wie dem traditionellen Lichtmodell werden dabei Kanäle angelegt für Diffuse, Specular oder Normal Maps, welche die Oberfläche des Meshes gestalten. Allerdings wird dabei nicht nur das Modell anhand dieser Parameter, sondern ebenfalls an den Objekten um es herum abgebildet.¹²⁵ Je nach Lichtstimmung und Umgebung reflektiert das Objekt beispielsweise in einer anderen Farbe.

¹²⁵ Learn OpenGL (o.J.)[B]



Abbildung 11: Einwirken der Umgebung auf das Mesh unter Verwendung von PBR

Zudem gilt das Gesetz der Energieerhaltung; fällt eine bestimmte Menge Licht auf ein Objekt, kann dieses nur so viel Licht wieder abstrahlen, wie von der Lichtquelle abgegeben. Diese Regel greift außerdem bei mehreren Objekten untereinander: Fällt dieselbe Menge Licht nun auf denselben Gegenstand, passiert jedoch zuerst eine Glasscheibe, absorbiert diese Scheibe zunächst einen Teil des einfallenden Lichtes. Das angestrahlte Objekt dahinter kann daher nur noch so viel Licht abstrahlen, wie nach dem Eintritt durch die Glasscheibe noch übrig ist. Im traditionellen Rendering-Modell wäre dieser Lichtverlust nicht berücksichtigt worden. Im selben Zug werden zudem Diffuse Maps im Gegensatz zu ihrer herkömmlichen Handhabung oft frei von jeglicher Ausleuchtung erstellt. Standardmäßig würde die Diffuse Map eines Steins bereits Schatten in dessen Furchen oder Löchern aufweisen, während beim PBR Workflow lediglich das steinige Material durch eine monochrome Farbe dargestellt wird, in diesem Fall ein konkreter Grauwert.¹²⁶ Auf den ersten Blick erscheint dieser Umstand eher Realitäts-schmälernd, im Endresultat verhält sich die Oberfläche des Objektes jedoch authentischer, da es nicht durch in der Diffuse Map vorgefertigte Lichtverhältnisse beschränkt wird.

Durch PBR erreichen Entwickler zwar eine neue Stufe auf dem Weg zum Realismus, allerdings ist der wirkliche Fotorealismus noch nicht erreicht. Je detaillierter und realistischer Game Assets kreiert werden, desto mehr Fehler fallen dem Betrachter auf. Un-

¹²⁶ Seymour (2013)

canny Valley wird dieses Phänomen bezeichnet und bezieht sich besonders auf die Abbildung von Menschen.¹²⁷ Neben Augen und Haaren stellt sich besonders die Haut als größte Baustelle der Entwickler heraus. Sie ist einer der überzeugendsten Aspekte einer Person und weist viele filigrane Details wie Poren, Härchen und Hautirritationen, dessen visuelle Darstellung neben hoher Rechenleistung auch dafür ausgelegte Grafik Engines fordert. Der Knackpunkt liegt vor allem in der Simulation komplexer Materialien, bei denen nicht nur Shader, sondern vor allem hochaufgelöste Texturen eine immense Rolle spielen. Diese müssten dann von Objekten aus der realen Welt eingescannt werden, wofür es versierte Fachkundige benötigt.¹²⁸

4.3.2 Stilismus durch Texturen

Entweder verfolgt das Spiel den realitätsnahen Ansatz oder entfernt sich davon, so dass es keinen „halben“ Realismus geben kann. Ohne den Anspruch auf realistische Grafik unterwirft es sich folglich einem Stil, weswegen dieser Ansatz als Stilismus bezeichnet werden kann. Einer dieser Grafikstile stellt das *Cel Shading* dar, bei welchem dreidimensionale Game Assets wie 2D Cartoon- oder Comicobjekte gestaltet werden. Merkmale dieses Typs sind vor allem helle, deckende Farben sowie breite, schwarze Konturen.¹²⁹

¹²⁷ Stuart (2015)

¹²⁸ Sweeney (2017), zit. n. Fischer (2017)

¹²⁹ Orland et. al. (2007), S. 19

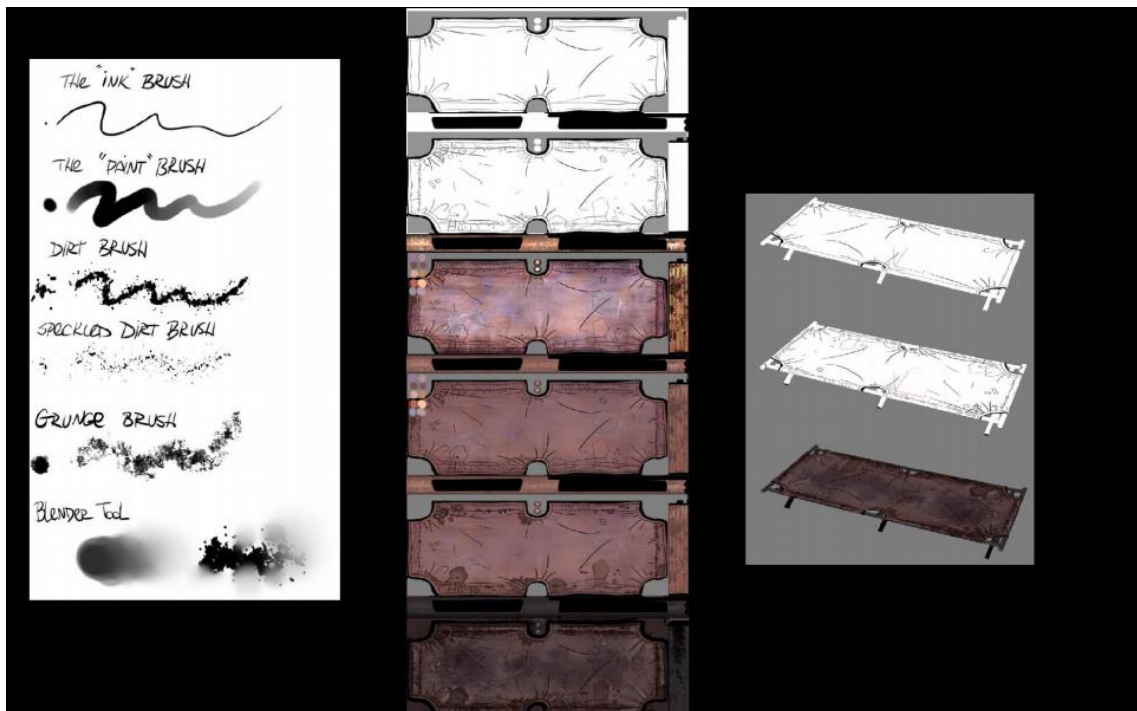


Abbildung 12: Erstellung der Diffuse Map eines Game Assets aus *Borderlands*

Die Texturen entstehen bei diesem Stil nicht aus Fotomaterial, sondern teilweise per Handzeichnung. Dabei werden die Formen zunächst manuell vorgefertigt, anschließend eingescannt und letztendlich in Zeichenprogrammen Schritt für Schritt koloriert. In der Nachbearbeitung wird außerdem eine stärkere Hervorhebung der Konturen hinzugefügt.¹³⁰ Viele der Assets erhielten zudem weder eine Normal Map, noch eine Specular Map.¹³¹

Vertreten ist eine Variante dieses Stils in der Shooter-Reihe *Borderlands* (*Borderlands*, *Borderlands 2*, *Borderlands: The Pre-Sequel*), obgleich besonders die Entwicklung des Grafikkonzepts die Vorteile des Stilismus widerspiegelt. Zu Beginn der Entstehung von *Borderlands* war das Design als realistisches Sci-Fi Setting angedacht. Allerdings fühlte sich das Gameplay zu extrem an – das Spiel zeichnete sich durch gigantische, verrückte Gegner, eine übermenschliche Sprunghöhe und übertriebene Effekte aus - und passte somit nicht zur den realistischen Ausrichtung. Aus diesem Grund wechselte der Stil zu einer comichaften Gestaltung.

¹³⁰ Gearbox, zit. n. Sterling (2009)

¹³¹ Orland et. al. (2010)



Abbildung 13: Wechsel von realistischer Grafik zu Comicstil in Borderlands

Der stilistische Ansatz offeriert genau die Aspekte der Gestaltung, die auch ein Maler zur Verfügung gestellt bekommt – Freiheit in der Farbauswahl und Proportionsgestaltung, mit der sich vor allem eine fiktive Spielwelt besser definieren lässt, als wenn sie versucht so realistisch wie möglich zu wirken. Einerseits finden sich auf diese Weise leichter Erklärungen für die Elemente des *Gameplays*. Wohin verschwinden die ganzen Objekte, die der Spieler aufnehmen kann? Wieso erhöht sich die Nachladegeschwindigkeit der Figur, wenn sie einen Level aufgestiegen ist? In der Realität wäre dies schwerer glaubwürdig zu vermitteln, denn beispielsweise auch der stärkste Mann kann keine 20 verschiedenen Waffen gleichzeitig tragen, ohne dass sie an ihm sichtbar zu erkennen sind. Entfallen Game Assets aufgrund ihrer Gestaltung sowieso aus der natürlichen Erscheinung, ist es auch vertretbar, dass eine Spielfigur fünf Meter aus dem Stand springen kann. Andererseits bekommt Farbe einen viel höheren Stellenwert zugesprochen, weswegen eine einzelne Diffuse Map die Chance erhält, Narration durch ihre Symbolkraft zu vermitteln. Der Stilismus hat demnach aus denselben Gründen, die für ihn sprechen, auch den Nachteil, dass besonders in einer Spielereihe der Darstellungshorizont beschränkt ist. An die Richtlinien, die den Stil erst definieren, muss sich durchgängig gehalten werden, andererseits fällt der Titel aus dem Muster und der Kontinuität der anderen Teile. Allerdings kann dieser Preis auch nur gering sein, da ein hervorstechender Stil zu einem Alleinstellungsmerkmal des Videospiels wird und es somit einen deutlicheren Eindruck hinterlassen kann.

4.3.3 Stilisierter Realismus aus Texturen und Modeling-Stil

Eine Sonderstellung nehmen Texture Maps ein, wenn sie im konkreten Zusammenhang mit dem Modeling-Stil verwendet werden. Bisher wurde lediglich der Einfluss von Texturen auf das Design berücksichtigt und sie als realistisch oder stilistisch eingestuft. In den Beispielen zu den beiden Fällen passte zudem der Grad an Realismus oder Stilismus zu den jeweiligen Texturausrichtungen – das Spiel mit realistischem Anspruch nutzte ebenso natürliche Proportionen bei den Objekten sowie jenes mit stili-

sierten Texture Maps auch unproportionierte 3D Modelle nutzte. Beim stilisierten Realismus werden hinsichtlich beider Aspekte jedoch unterschiedliche Designkonzepte genutzt.

Eine Möglichkeit ist es, realistische Proportionen bei den 3D Objekten zu nutzen, während diese mit stilistischen Texturen versehen werden. Bei dieser Variante soll vor allem die Narration der Geschichte im Fokus stehen, die über die realen Geschehnisse hinaus auf einer tieferen, symbolhaltigen Ebene erzählt werden soll. Dabei bildet der Realismus der Modelle die Basis der Geschehnisse, vermittelt das Bekannte und Alltägliche, welches zu Beginn der Handlung noch vom Spieler unberührt vor ihm ausgebreitet wird. Währenddessen bedienen sich die Texturen der Vorteile des Stilismus und reflektieren die Ereignisse des Spiels. Zum einen erlauben sie Spielraum für Ungenauigkeiten und Fehler in Bezug auf die realistische Basis, von welchen der Spieler nicht abgelenkt wird, da er anhand des Designstils bereits erkennt, dass der Schauplatz nicht real ist.¹³² Zum anderen erzählen wiederkehrende Farbschemata und ausgewählte Farbakzente die Geschichte visuell parallel zu Animationen und Dialogen. Bei diesem Konzept kann der Designer auf die Regeln und Möglichkeiten der Farbenlehre zurückgreifen und seine Erzählung trotzdem einer in der Realität angesiedelten Welt stattfinden lassen. *Life is Strange* stellt solch ein Beispiel dar, wobei die Erlebnisse zweier Teenager in einem realistischen Lebensumfeld als „animierte Konzeptkunst“¹³³ erzählt werden.

¹³² Caplain, zit. in 80.lv (2016)

¹³³ Ebd.



Abbildung 14: Stilisierte Texturen auf realistischer Modeling Grundlage in Life is Strange

Die zweite Möglichkeit stilisierten Realismus zu verwenden, besteht darin, ein realistisches Texture Mapping auf der Grundlage von stilisiertem Modeling zu erstellen. In diesem Fall geht der Spieler aufgrund der realistisch designten Texturen bereits zu Beginn davon aus, dass sich die Handlung des Spiels in der realen Welt abspielt. Hinsichtlich dem Rendering und der Simulation von Materialien werden daher ebenso hohe Ansprüche gestellt, wie bei durchgehend realitätsnahen Videospielen. Allerdings weisen die unrealistischen Proportionen der 3D Modelle auf etwas Unnatürliches, Unbekanntes und somit auch Bedrohliches hin, welchem sich der Spieler stellen muss. Auf diese Weise erhalten die Entwickler die Chance, eine abstrakte Geschichte in der Realität zu verankern und ihr etwas Natürliches und Menschliches zu verleihen. Beispielsweise greift *Rocksteady's Batman* Trilogie diesen Ansatz auf, wodurch es den Entwicklern gelungen ist, die surrealen Helden und Bösewichte einer Comicbuchreihe in eine lebendige, realistische Spielwelt zu transferieren.



Abbildung 15: Realistische Texturen auf Basis stilisiertem Modelings in Batman: Arkham Asylum

Trotz dass Batman's Stadt Gotham eine in der Realität funktionierende Welt repräsentiert, findet er sich von theatralischen und verrückten Bösewichten umgeben. Die hyperrealistischen Proportionen sowie übermenschliche Größen der Gegenspieler verbreiten diesen Wahnsinn, sind jedoch immer noch als Menschen zu erkennen. Hierfür sorgen vor allem detaillierte, hochaufgelöste Texturen und Physically-based Rendering.¹³⁴ Dieser Trend setzt sich vor allem in den Einrichtungen der Irrenanstalt „Arkham Asylum“ fort; unnatürlich hohe Wände und verwinkelte, unregelmäßig verlaufende Mauerwerkselemente erzeugen dunkle Schatten, in denen der Spieler die nächste Gefahr vermuten kann.¹³⁵

4.4 Procedural Textures

Der Spagat zwischen grafischer und narrativer Qualität bis hin zu einer flüssigen Performanz ist nicht immer leicht, denn fortwährend lässt der technische Fortschritt Entwickler eine neue Ebene von Videospielen erreichen; ihre Welten wachsen an Größe und Detailgrad, welche es mit einer Fülle an Assets auszustatten gilt. Daher kann irgendwann der Zeitpunkt kommen, an welchem die Vorteile von herkömmlichen Texture Maps nicht ausreichen, um dieser Nachfrage gerecht zu werden. In einer Zeit, wo eine solche Massen an verschiedensten Objekten mit unterschiedlich zusammengesetzten

¹³⁴ Wallace (2015), S. 29-31

¹³⁵ Ebd., S. 59

Materialien nötig sind, erfordert es eine dynamischere Generierung und einen modularen Einsatz von Texturen – durch prozedurale Texturen.¹³⁶

4.4.1 Definition und Wirkungsumfeld von Procedural Textures

Eine *Procedural Texture* stellt für sich kein Bitmap Image im eigentlichen Sinne dar; vielmehr beschreibt sie einen Algorithmus, mit welchem eine Texture Map mathematisch durch verschiedene Parameter synthetisiert werden kann. Dabei referiert er auf eine Farbinformation, welche durch eine Abfrage der X- und Y-Koordinate abgerufen werden kann. In diesem Zusammenhang wird der entsprechende Farbwert nicht in einer Datei gespeichert, sondern durch den Code generiert. Dadurch wird außerdem der Bereich zwischen den einzelnen Pixeln gefüllt, welcher bei Standard Texture Maps ansonsten nicht definiert ist.¹³⁷ Insofern, dass digital erzeugter Inhalt in jedwedem Fall durch Programmcode entsteht, ist es zunächst notwendig, Procedural Textures als die für das Wirkungsumfeld der Spieleentwicklung angewandte Begrifflichkeit abzugrenzen.

Hinsichtlich ihrer Funktionsweise können im Allgemeinen explizite und implizite prozedurale Texturen unterschieden werden. Während beim expliziten Verfahren Koordinaten direkt in einer determinierten Reihenfolge in die Textur gezeichnet werden, wird die Map bei der impliziten Arbeitsweise durch Abfragen bezüglich ausgewählter Punkte erstellt. Wird der Umstand betrachtet, dass Texturen auf dreidimensionalen Objekten zunächst gerendert werden müssen, wird ersichtlich, dass die implizite Verfahrensweise den Bedürfnissen der Spieleentwicklung eher entspricht. Ein Renderer in der Game Engine bewertet die eingelesene Textur nach einer festgeschriebenen Reihenfolge, die sich jedoch nicht an dem Code der Procedural Texture, sondern an seinem eigenen Algorithmus orientiert. Da implizite prozedurale Texturen Output in einer Queue ausgeben, passt dieses Verarbeitungsmuster zur Funktionsweise des Renderers.¹³⁸

4.4.2 Potenzial von Procedural Textures

Aufgrund der Problematik des steigenden Fassungsvermögens von Spielwelten stehen procedural Texture Maps immer öfter im Diskurs. Große Spieletitel wie *Horizon Zero Dawn* oder *Uncharted 4* griffen im Zuge dessen bereits auf diese Art des Texturings zurück.¹³⁹ Besonders in Hinsicht auf die Vielzahl an Game Assets, welche für solch ein

¹³⁶ Nogue (2016)

¹³⁷ McCombs (2017)

¹³⁸ Ebert et. al.. (2003), S. 12-13

¹³⁹ Allegorithmic (2017)[B]

Videospiel zur Verfügung stehen müssen, kristallisieren sich prägnante Vorteile von prozeduralen Texturen heraus.

Procedural Textures weisen zunächst im Vergleich zu den speicherintensiven Bitmap Texturen eine höhere Komprimierung auf; während bei herkömmlichen die Dateigröße Texture Maps oft in Megabytes gemessen werden, verbrauchen procedural Textures lediglich Kilobytes. Es wird somit eine beachtliche Menge an Speicherkapazität gespart, was entweder der Performanz des Spiels zu Gute kommt oder mehr Platz für weitere Texturen und damit auch Detailgrad schafft. Prozedurale Texture Maps besitzen weiterhin keine Restriktion bezüglich ihrer Auflösung. Nähert sich der Spieler beispielsweise einem 3D Objekt, verliert dessen Oberfläche nicht an Schärfe im Gegensatz zu Standard Textures. Unter Berücksichtigung dieses Gesichtspunktes lässt sich zudem nachvollziehen, dass dieser Typus an keine spezifische Geometrie gebunden ist und somit form für jedes beliebige Objekt sein kann. Oberflächenmaterialien können folglich nun global angelegt werden, wodurch sie an verschiedenen Modellen angewendet werden können. In Anbetracht an die herkömmliche Arbeitsweise, nur an ein bestimmtes Objekt adaptierbare Texturen zu designen, erlaubt dieses Verfahren nun die Wiederverwertbarkeit von Texture Maps. Ihre code-basierte Natur ermöglicht außerdem eine Parametrisierung, sodass eine Klasse entsteht, welche durch variable Einstellung mehrere sich ähnelnde Unterarten der Textur ausbilden kann.¹⁴⁰ Es kann so beispielsweise ein Material erstellt werden, welches einen Vinylboden repräsentiert und obgleich die visuellen Merkmale, die diesen Typ Boden als solchen identifizieren, erhalten bleiben, können ausgehend davon mehrere Texturen für Vinylböden erzeugt werden. Mittels Justieren ausgewählter Parameter ist es infolgedessen möglich, den Boden nicht nur bezüglich seiner grundlegenden Farbe zu manipulieren, sondern ebenfalls hinsichtlich eventueller Kratzer, Glanzpunkte oder Erhöhungen. Diese Feineinstellungen können dabei effizienter durchgeführt werden als an in Auflösung und Größe determinierten Bitmap Texture Maps, bei welchen nach Abschluss der Arbeit an der Textur nur durch komplizierte Manipulationen Änderungen getätigt werden können.¹⁴¹ Dieser Ansatz kann des Weiteren mit dem des Physically-based Shading kombiniert werden; anstelle von Texture Maps können für die Kanäle des PBR-Shaders alternativ auch procedural Textures eingesetzt werden.

Diese Vielfalt an Möglichkeiten beim Texturieren gestaltet sich für Designer besonders attraktiv, jedoch benötigt sie im Vorfeld wie während der Erarbeitung der Textur aufgrund ihrer codischen Charakteristik programmatisches Knowhow. Das Schreiben der Anweisungen, um die entsprechenden Muster darstellen zu können, kann teilweise einen höheren Aufwand bedeuten, als dass überhaupt eine schwerwiegende Veränderung an der resultierenden Textur sichtbar wird. Da solche Algorithmen mit Zufallsfak-

¹⁴⁰ Ebert et. al. (2003), S. 14

¹⁴¹ Pettit (2015)[A]

toren arbeiten, um die unterschiedliche Oberflächenausprägungen zu synthetisieren, ist der Output allerdings nicht vorhersehbar wie bei einer standardmäßigen Standard Texture. Dies kann einerseits als Stärke angesehen werden, wenn eine realistische Nachbildung von Materialien angestrebt, obgleich in der Natur ebenfalls keine allumfassende Regelmäßigkeit herrscht. Andererseits kann sich diese nur eingeschränkte Kontrolle auch als Schwäche erweisen, sobald bestimmte Aspekte eines Materials an konkrete Restriktionen ausgebildet werden müssen.¹⁴²

Diese Grenzen referieren vor allem auf die manuelle Arbeit mit procedural Textures. Um die Vorteile allerdings auch als Designer ohne tieferreichende Programmierkenntnisse weitestgehend nutzen zu können, erfordert es eine Software, welche in der Lage ist, mittels einer benutzerfreundlichen grafischen Oberflächen Designern das Potenzial von prozeduralen Texturen ausschöpfen zu lassen. Derzeit stehen diverse solcher Anwendungen auf dem Markt zur Verfügung; darunter zum Beispiel *Filter Forge*, *MapZone* oder *Allegorithmic Substance Designer*.¹⁴³ Besonders letzterer hat sich heutzutage in der Branche etabliert, indem er bei einigen namenhaften Spieletiteln wie *Horizon Zero Dawn*, *Uncharted 4: A Thief's End*, *Prey* oder *Ghost Recon: Wildlands* sowie vielen weiteren zum Einsatz kam.¹⁴⁴ Die Software stellt unter anderem Tools zur Verfügung, um 3D Objekte zu texturieren, rendern und die Resultate dieser Prozesse so vorzubereiten, dass sie beispielsweise anschließend in einer Game Engine verwendet werden können. Dabei steht es dem Nutzer zusätzlich frei, Materialien von Grund auf neu anzufertigen oder sogenannte *Presets*, in diesem Fall vorgefertigte Textursammlungen oder Shader, zu verwenden und diese den eigenen Bedürfnissen anzupassen. Während Substance Painter nicht nur die professionelle und qualitativ hochwertige Werkzeuge für das Texture Design offeriert, bietet das Programm eine für den Designer, der sich und seine Mitarbeiter permanent mit dem optischen Feedback seiner Arbeitsschritte konfrontieren muss, komfortable Funktion; ein *real-time PBR viewport* zeigt direkt die generierten Texturen am Modell so, wie sie letztendlich im Spiel selbst aussehen würden und aktualisiert diese bei Änderungen in Echtzeit.¹⁴⁵

¹⁴² Ebert et. al. (2003), S. 14-15

¹⁴³ Slant (2017)

¹⁴⁴ Allegorithmic (2017)[B]

¹⁴⁵ Ebd. [C]



Abbildung 16: Viewport in Allegorithmic Substance Painter

Dieses Feature ersetzt einen umständlichen *Workflow*, bei welchem der Designer immer wieder zwischen dem Grafikbearbeitungsprogramm, in welchem er an der Textur arbeitet, und dem 3D Programm beziehungsweise der Engine, wo die Texturen erst als Material akkumuliert und auf dem Mesh angezeigt werden, wechseln musste. Die Echtzeitdarstellung fällt unter Allegorithmic's *Dynamic Material Layering*, was dem Nutzer zusätzlich automatisch ein benutzerdefiniertes Interface zusammenstellt, welches es ihm nicht nur erlaubt, so dynamisch wie möglich mit den verschiedenen Tools zu hantieren, sondern ebenfalls den Überblick über verwendete Ressourcen und deren Rahmenbedingungen nicht zu verlieren.¹⁴⁶

Folglich stellen procedural Textures einerseits eine notwendige Alternative zur herkömmlichen Arbeit mit Standard Texture Maps dar, da es eine performantere und effizienter nutzbare Methode der Generierung von Texturen erfordert, um die hohe Nachfrage nach qualitativen Game Assets für immer größere werdende Spielwelten decken zu können. Solche Algorithmen können daher zu einem gewissen Teil die umständliche Handarbeit der Designer erleichtern, jedoch erfordert es andererseits weiterhin Menschen, die nicht nur den kreativen Input mit sich bringen, sondern ihn auch mit Know-how zu einem im Spiel nutzbaren Gesamterzeugnis transferieren können. Dabei stellen Anwendungen wie Allegorithmic Substance Painter eine nutzerfreundliche Schnittstelle zwischen Designer und den procedural Textures dar und gewährleis-

¹⁴⁶ Ebd.

ten dadurch das Freilegen all der Eigenschaften dieser Texturen, von welchem das Videospiel letztendlich profitieren kann.

5 Analyse des Texture Mappings am Beispiel von *Tomb Raider's Lara Croft*

5.1 Vorstellung der Spielereihe

Sein Debut feierte das *Tomb Raider Franchise* am 31.10.1996 unter dem Titel *Tomb Raider (featuring Lara Croft)*, welchem bis ins Jahr 2016 17 weitere Teile für verschiedene Plattformen folgten, darunter PC, PlayStation, PSP, Sega, Nintendo Wii, Xbox und Android.¹⁴⁷ Vormalig als satirisches weibliches Double von Abenteurerfigur India Jones angedachte „Laura Cruz“ entwickelte sich während der Entwicklung des ersten *Tomb Raider* zu „Lara Croft“, eine der populärsten Videospielheldinnen, deren Bekanntheit sich sogar über die Branche hinaus verbreiten konnte. Nicht nur durch verschiedene Adaptionen in Romanen, Comics und Filmen, sondern vor allem durch die vielen Videospielteile selbst, wurde die Geschichte der Protagonistin immer mehr verfeinert oder verändert. Allen zu Grunde liegt jedoch Lara Croft, eine Archäologin und Schatzsucherin, die auf der ganzen Welt Höhlen, Ruinen und Grabstätte erforscht und dadurch oft in ein kämpferisches und action-geladenes Abenteuer gerät. Ursprünglich wurde sie dabei von Toby Gard geschaffen, welcher als Designer beim britischen Entwicklerstudio *Core Design* mitwirkte.¹⁴⁸ Das Studio war von der Schöpfung der Serie bis *Tomb Raider: The Angel of Darkness* in 2003 für die Entwicklung zuständig; anschließend wurde es durch *Crystal Dynamics* abgelöst, welches bis *Rise of The Tomb Raider* im Jahr 2016 für die Umsetzung der *Tomb Raider* Reihe verantwortlich war.¹⁴⁹

5.2 Charakteristik des Models „Lara Croft“

Die Figur Lara Croft beschreibt eine junge Frau, schätzungsweise in den Mittzwanzigern, mit langen, braunen Haaren, welche sie zu einem Pferdeschwanz gebunden trägt. Sie ist circa 1,75m groß und zeichnet sich durch eine schlanke, feminine Figur aus.¹⁵⁰ Ihr typisches Outfit besteht zumeist aus Shorts oder einer langen Hose, Stiefeln und einem eng anliegenden Tank Top und ist in ein Farbschema aus Blau- und Brauntönen eingebettet. Lara sieht sich in ihren Abenteuern oftmals verschiedenen Gegner entgegengestellt, welche sie auf unterschiedlichen Wegen besiegen kann. Neben anderen Waffentypen sind dabei vor allem zwei Pistolen ihr Lieblingsaccessoire und Markenzeichen.

¹⁴⁷ Tombraider-Games.de (o.J.)[A]

¹⁴⁸ Tomb Raider Wiki (o.J.)[A]

¹⁴⁹ Tombraider-Games.de (o.J.)[A]

¹⁵⁰ Tombraider-Games.de (o.J.)[B]

Das Model der Protagonistin entwickelte sich im Laufe der Zeit einerseits parallel zum Stand der Technik sowie andererseits durch den Wechsel von Core Design auf Crystal Dynamics. Der nun stetig höher angesetzte Polycount und immer hochauflösendere Texturen beeinflussten zunehmend Laras Erscheinungsbild, welches sich somit über drei verschiedene Zeitachsen veränderte. Die drei Zeitschienen (Titel, die Handheld-Konsolen und Mobiltelefon erschienen sind werden dabei außer Acht gelassen) repräsentieren je einen unterschiedlichen Ansatz dafür, welche Abenteuer Lara Croft erlebt und welche Ereignisse in ihrem Leben sie zu ihrem Hobby brachten.

5.2.1 „Original“ Zeitachse

Die anfänglichen Titel verweisen auf eine Lara, die zunächst Reichtum und Luxus genoss, da sie als Kind einer britischen Aristokratenfamilie zur Welt kam und dadurch in dieses Leben geboren wurde. Dies ändert sich allerdings, nachdem sie mit 21 Jahren nur knapp einen Flugzeugabsturz überlebte und sich infolgedessen allein in der Wildnis beweisen und überleben musste. Dieses Erlebnis prägte sie selbst nach ihrer Rückkehr nach Hause so sehr, dass sie die Gesellschaft der Wohlhabenden als langweilig befand und sich seit dem der Schatzsuche und Grabforschung widmete, welche sie sich durch das Schreiben von Romanen über ihre Erlebnisse finanzierte.¹⁵¹

In dieser Zeit dominiert zunächst das ursprüngliche Lara Croft Outfit. Das Tank Top weist fast durchgehend ein helles blau auf, die kurzen Shorts sind braun bis khaki-braun gefärbt. Lara trägt einen auffallend dicken Gürtel mit einer großen Metallschnalle, dunkelbraune Handschuhe mit Aussparungen für die Finger sowie zwei Waffengürtel für die Zwillingspistolen, welche an ihren Oberschenkeln festgemacht sind. Herausstechendes Merkmal dieser Epoche stellt vor allem die überdimensionierte Oberweite des Modells dar, eingerahmt durch die kurzen Riemen eines kleinen Rucksacks, welchen sie auf dem Rücken trägt.

5.2.2 „Legend“ Zeitachse

In dieser Version von *Tomb Raider* gehört Laras Familie ebenfalls der Aristokratie an, allerdings ist Laras Vater, Lord Croft, von Beruf Archäologe, was auch in seiner Tochter Begeisterung für dieses Hobby weckte. Der Flugzeugabsturz findet auch in dieser Zeitschiene statt, obgleich sich die Protagonistin im Alter von neun Jahren befindet und mit ihrer Mutter zusammen im Familienjet verunglückt. Die folgenden Ereignisse in den Ruinen um die Absturzstelle führten zum Verschwinden von Lady Croft, für welches

¹⁵¹ Tomb Raider Wiki (o.J.)[C]

sich die junge Lara verantwortlich macht und von diesem Tag an trainiert und lernt, um die Umstände des Verlusts ihrer Mutter aufzudecken.¹⁵²

Die Wende zur Legend-Zeitachse spiegelt sich ebenfalls in Laras Erscheinungsbild wieder. Neben einem schon deutlich reduzierteren Brustumfang fällt besonders der von Teil zu Teil wechselnde Stil des Oberteils auf. Bereits in *Tomb Raider: Legend* entschied sich die Design Leitung vom bekannten blauen Tank Top hin zu einem braunen T-Shirt. In *Tomb Raider: Anniversary* fand man sich jedoch zum anfänglichen Konzept des Tank Tops zurück. Allerdings erhielt es in *Tomb Raider: Underworld* eine weitere Überarbeitung und wies neben einer Abwandlung des Schnitts statt hellblau eine dunkelbraune Färbung auf.

5.2.3 „Survivor“ Zeitachse

Die letzte Zeitachse geht von einer durch die Beziehung zu ihrem Vater animierte Lara Croft aus. Nachdem Lady Croft auch hier bei einem Flugzeugabsturz ums Leben kommt, bleibt die kleine Lara mit ihrem Vater allein zurück. Dessen Forschungsarbeit und archäologischen Expeditionen lassen auch auf Lara den Funken für dieses Hobby überspringen. Von dem Tod seiner Frau gebrochen, vergräbt sich Lord Croft jedoch in seine Arbeit und der Suche nach der Unsterblichkeit, deren Geheimnis er jedoch nie herausfindet, sodass er schließlich Selbstmord begeht. Lara beschreitet trotz dessen den Pfad einer Schatzsucherin und versucht das Erbe ihres Vaters zu wahren und weiterzuführen.¹⁵³

Der inzwischen bahnbrechende Zuwachs an Möglichkeiten bezüglich Grafik hinterließ deutliche Spuren an Lara Croft, deren Proportionen und Texturen nun im realistischen Design erscheinen. Zusätzlich werden die kurze Hose gegen eine lange sowie der Rucksack gegen eine Art Waffengürtel, der wie eine Tasche über die Schulter getragen wird, ausgetauscht. Zudem erhält Lara ein neues Accessoire – eine Lederkette um den Hals, an welcher ein Jade-Anhänger befestigt ist. Die Besonderheit der Teile der Survivor Zeitschiene stellt vor allem das Kleidungssystem dar, in welchem der Spieler zwischen durch die Geschichte beziehungsweise durch Freischaltcodes verfügbaren unterschiedlichen Outfits wählen kann.

¹⁵² Tomb Raider Wiki (o.J.)[B]

¹⁵³ Tomb Raider Wiki (o.J.)[D]

5.3 Analyse des Texture Mappings

5.3.1 Texturing in der “Original” Zeitachse



Abbildung 17: Zusammenschnitt von Referenzbildern zu Tomb Raider: Festuring Lara Croft

Das Texturing der ersten Lara Croft war vor allem dramatisch durch die Limitation des damaligen Stands der Computertechnik restringiert. Zur Anwendung kam lediglich eine Diffuse Map, welche Grundfarbe sowie erste Ansätze von Schattenbildung aufwies. Die Shorts erscheinen durch das Anzeigen von Falten besonders im Schritt knapp und eng, so wie auch Laras Oberteil durch die aufgemalten Abzeichnungen von Bauchmuskulatur und Brust ebenfalls am Körper anliegt. Im Gegensatz dazu lassen die Handschuhe und Waffengürtel durch ihre dunkelbraune Färbung kaum Detail erahnen; einzig die Gürtelschnalle sticht durch einen durch Farbe erzeugtes Goldschimmern hervor und fokussiert den Blick infolgedessen auf die schmale Taille Laras. Rucksack sowie Stiefel zeichnen sich nur durch einen monochromen Branton aus, durchbrochen von jeweils profanen Schnallen und der Andeutung von weißen Socken sowie Schnürsenkeln, die farblich mit den rotbraunen Haaren und Lippen des Models synchronisieren. Letztere sind ausschließlich auf der Diffuse Map erzeugt – ebenso wie die Augenpartie und die Nase - und auffallend voll. Der Pferdeschwanz zeigt ein gesträhtes Muster aus einem helleren und einem dunkleren Branton auf, ist jedoch trotzdem bereits als Haar zu erkennen, welches durch ein gelbliches Band zusammengehalten wird. Haut ist zu diesem Zeitpunkt lediglich durch ihren cremefarbenen Ton zu erkennen. Auf die beiden Pistolen lassen zunächst neben ihrer Form und der Art, wie Lara sie verwendet, Grautöne mit einzelnen dunkleren Linien schließen, welche Lauf, Abzug und Mündung optisch voneinander trennen.



Abbildung 18: Zusammenschnitt von Referenzbildern zu *Tomb Raider II: Starring Lara Croft* und *Tomb Raider III: Adventures of Lara Croft*

In den beiden folgenden Teilen *Tomb Raider II: Starring Lara Croft* sowie *Tomb Raider III: Adventures of Lara Croft* kommt es bereits zu ersten Model-technischen Verbesserungen, indem der Pferdeschwanz länger und mit einem nach oben stehenden Haaransatz ausgestattet wird und die Oberweite abgerundeter wirkt. Diese Überarbeitung erstreckt sich insofern auf die Diffus Map aus, dass die Falten der Kleidung durch den Wechseln von helleren und dunkleren Highlights stärker definiert dargestellt sind. Auch an den Waffenhalterungen, Pistolen und Socken sind nun erste Schattierungen zu erkennen.



Abbildung 19: Zusammenschnitt von Referenzbildern zu *Tomb Raider: The Last Revelation* und *Tomb Raider: Chronicles*

Mit *Tomb Raider: The Last Revelation* und *Tomb Raider: Chronicles* wird der Fokus auf Laras Weiblichkeit weiterhin ausgebaut, indem das Top tiefer ausgeschnitten dargestellt wird und die anfangs deutlich sichtbare Bauchmuskulatur nun nicht mehr zu sehen ist. Besondere Betonung liegt nun zudem auf dem Hintern, welcher durch verstärkte und feinere Schattierung als in den Teilen zuvor mehr herausgehoben wird. Allerdings steuert dieser Ansatz damit ebenfalls auf eine detailliertere Visualisierung der Haut hin; Glanzpunkte sowie sichtbare Schatten im Dekolleté und an den Schlüsselbeinen zeugen vom ersten Versuch mehr Menschlichkeit umzusetzen, welche nun auch im Gesichtsbereich durch kleine Helligkeitsakzente an Mund und Nase greift. Auch das Material des Rucksacks lässt nun mehr an Textilien erinnern, da geschwungene, Falten repräsentierende Ausschattungen sowie Highlights an Lasche und Schnallen nun auf das Verhalten von Stoff deuten.

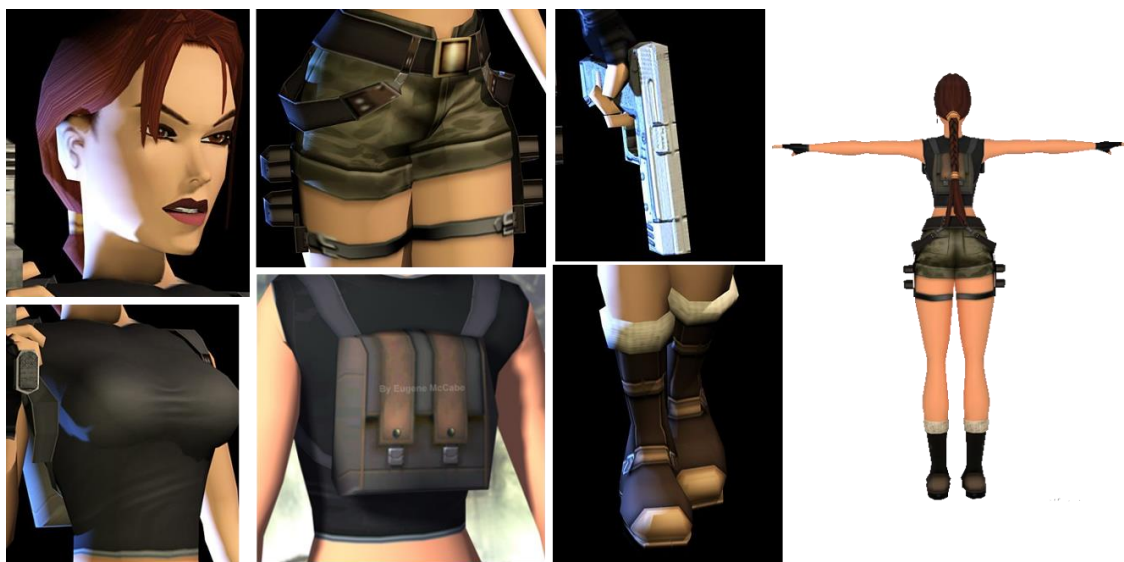


Abbildung 20: Zusammenschnitt von Referenzbildern zu Tomb Raider: Angel of Darkness

Obwohl *Tomb Raider: Angel of Darkness* noch zur „Original“-Zeitschiene gehört, symbolisiert dieser Teil bereits den Umschwung zur neuen Achse. Einerseits ist ein deutlich höher angesetzter Polycount sichtbar, wodurch die Waffengürtel zum Teil durch wirkliche Geometrie abgebildet werden und der Pferdeschwanz durch den hinzugefügten Mittelscheitel-Pony deutlicher Haare erkennen lässt. Zudem entwickelten sich Laras Hände, welche zuvor nur durch einen Quader repräsentiert wurden, zu gemodelten Fingern. Das Texturing nähert sich währenddessen weiter einer realistischeren Darstellung an. Die Kleidung weist nun feinere Falten auf, welche auch in diesem Fall wieder Laras weibliche Konturen betonen. Allerdings erscheint das Top in der Diffuse Map als hochgeschlossen und dunkelgrau statt hellblau, wodurch Lara etwas mehr Seriosität verliehen wird. Die Shorts zeugen diesmal von einem Oliveton, was in Kombination mit dem Oberteil und dem nun aus einem festeren Material als Stoff zu bestehenden Rucksack einen militärisch angehauchten Eindruck vermittelt. Erstmals sind auch Verschlüsse, Nieten und Haken an den Waffengürteln festzustellen sowie auch die zierlose Metallplatte des Hüftgürtels nun schon als richtige Schnalle zu erkennen ist. Auch das Material der Pistolen scheint deutlich metallischer durch mehr Glanzhighlights und mechanisches Detail in ihrem Aufbau. Laras Gesicht erfährt zusätzliche Feinheit, wodurch eine ausgeprägtere Kopfform entsteht. Augen, Nase, Ohren und Mund wirken weniger aufgesetzt als in den bisherigen Titeln und fügen sich dezenter und weicher in das Gesicht ein. Allerdings wird all dies immer noch lediglich in eine Diffuse Map abgebildet.

Das Texturing der „Original“ Zeitachse zielt besonders auf die übermäßige Betonung der Weiblichkeit des Models ab. Weibliche Hauptfiguren stellten in dieser Phase von Videospielen eine Seltenheit dar, weshalb dieses Designkonzept angewandt wurde, um das Spiel für männliche Spieler attraktiver zu gestalten. Die wenigen Möglichkeiten der Tomb Raider Spiel-Engine flossen daher eher in Lara als Sexsymbol als in ihren Status als Archäologin, was besonders in Hinsicht auf weibliche Spieler Debatten aus-

löste. Mit dem Wechsel in *Tomb Raider: Angel of Darkness* auf die neue Grafikengine von Core Design war es den Entwicklern anschließend möglich, Lara nicht nur visuell zu optimieren, sondern auch einige farbliche und modeltechnische Veränderungen vorzunehmen. Auch wenn Lara Crofts Proportionen längst nicht natürlich für eine athletische junge Frau erschienen, stimmten das abgedämpfte Farbkonzept und die kleine Reduzierung des Brustumfangs die Spielerinnen milder.

5.3.2 Texturing in der „Legend“ Zeitachse



Abbildung 21: Zusammenschnitt von Referenzbildern zu Tomb Raider: Legend

In *Tomb Raider: Legend* setzt sich der in der bisherigen Zeitschiene begonnene Trend fort. Obwohl Laras Körpermaße noch etwas hyperrealistisch dargestellt wurden, beginnt sich das Texturing auf ihre sichtbare Athletik zu richten. Der für die Haut ausgelegte Bereich in der Diffuse Map weist Ansätze für die Muskulatur auf, so wirken Laras Knie, Arme und Bauch sehniger und sportlicher. Der Pferdeschwanz zeigt eine deutlichere Haarstruktur mit einem fransigen Pony in der Stirn auf, was der Protagonistin ein weniger statisches Aussehen verleiht. Die Haarfarbe wurde entsättigt und schimmert nun eher in einem Kupferton als einem satten rot. Der militärische Kleidungsstil wird besonders hinsichtlich der Stiefel und des Rucksacks – beides in fester Lederoptik mit vielen silbernen Riemen und Schnallen – fortgesetzt, erhält jedoch durch das Hinzufügen von Ausrüstungsgegenständen am Gürtel mehr den Eindruck von Kletterausrüstung. An letzterem sind nun größere metallische Niete zu erkennen; außerdem zierte ein Muster die Gürtelschnalle, welche nun nicht mehr Gold, sondern Silberfarben ist.

Auch bei diesem Titel liegen Hose und Top eng am Körper an und sind sehr knapp, allerdings fällt auf, dass das Standard Oberteil ausnahmsweise ein T-Shirt mit längeren Ärmeln darstellt. Dieses lässt zwar das bekannte hellblau als Unterteil durchblicken, die dominante Farbe ist allerdings ein kühles hellbraun. Auch gibt es nur bedingt Faltenbildung im Dekolleté; alle anderen ranken sich um die Armbeugen sowie den unteren Saum des Oberteils. Laras weibliche Züge werden somit deutlich abgedämpft und wirken durch die Wahl des T-Shirts nicht mehr auf eine anzügliche Weise, sondern dezent im Kontrast zu der restlichen praktischen Ausrüstung. Die Zwillingspistolen erhielten zudem ebenfalls eine Überarbeitung, indem sie nun nicht mehr ausschließlich aus einem einzigen hellen, metallischen Material bestehen, sondern ebenfalls aus einem zweiten, dunkleren. Durch die Verwendung dieses Hell-Dunkel-Kontrasts erscheint das helle, im oberen Teil der Kanone angebrachte Metall leichter als das dunkle, welches Lauf, Abzug und Magazin bedeckt. Zusätzliches Detail erscheint auf dem Leichtmetall durch ein kleines rotes Symbol sowie feine Linien und Schrift. Die Farben der Diffuse Map nähern sich im Resultat in diesem Teil einem kühleren Farbkonzzept an.



Abbildung 22: Zusammenschnitt von Referenzbildern zu Tomb Raider: Anniversary

In *Tomb Raider: Anniversary* kehrt man erneut zu Laras klassischem Outfit zurück. Sie trägt wieder das hellblaue Tank Top, welches bis zum Gürtel reich und hochgeschlossen geschnitten ist. In dieser Version wurde mit Licht und Schatten sparsam gearbeitet, sodass ein leichter Anstieg oder Abfall von Helligkeit des Blautons auf die Beschaffenheit des Körpers schließen lässt. Zum ersten Mal in der Entwicklungsgeschichte der Reihe weist es nun eine geriffelte Oberfläche wie von Wolle oder Cord auf, welche sich auch bei den Socken durchsetzt. Die Shorts sind dagegen nur leicht schattiert. Bei genauerer Betrachtung der Stiefel fällt auf, dass sich das Farbschema von Schwarz und Silber zu Hell- und Dunkelbraun geändert hat und sie somit nicht mehr nach Kampfstie-

fein, sondern nach Schuhwerk zum Wandern oder Klettern aussehen. Diese Veränderung durchliefen ebenfalls Gürtel und Rucksack, obgleich bei letzterem erneut das Konzept eines stofflicheren Materials zum Einsatz kommt. Die Schnalle des Gürtels zeigt nunmehr eine goldene Färbung auf, ebenso wie Nähte und Nieten. Im Gegensatz dazu hebt sich das metallische Silber der Pistolen vom restlichen Design ab. Auch hier entfernt sich das Design von schwererer Ausrüstung und geht in die Richtung einer leichteren Waffe, an welcher lediglich der Griff – zu erkennen durch eine monochrom schwarze Einfärbung mit Nietenaufsätzen – aus einem anderen Material (wahrscheinlich Hartgummi) besteht. Lara Croft selbst zeigt nun stärker differenziertere Hautpartien, besonders was ihr Gesicht betrifft. Durch einen dunkleren Rosaton zeichnen sich die Wangenknochen vom Rest der Gesichtspartie ab, während der Bereich an Kinn, Nase und Augen durch einen dunkleren Hautton abgeschattet ist. Laras Gesicht wirkt somit im Ganzen natürlicher und definierter, ebenso wie der Pferdeschwanz, bei welchem nicht nur ein Flechtmuster erkennbar ist, sondern dessen Haarteile auch realistisch angeordnet sind.

Insgesamt erfährt das Designkonzept wieder mehr Sättigung und wirkt besonders durch die Wiederkehr des typischen Hellblaus wärmer. In Kombination mit dem Ersatz des Silbers in Laras Outfit mit Gold wird der militärische, kämpferische Aspekt ihrer Erscheinung gedämpft und bringt den Charakter als Archäologin mehr zur Geltung. Kleidung und Ausrüstung scheinen demnach eher auf Kletter- oder Wandertouren ausgelegt zu sein, da besonders auch Rucksack sowie Pistolen aus leichterem Material gefertigt zu sein scheinen.



Abbildung 23: Zusammenschnitt von Referenzbildern zu Tomb Raider: Underworld

Tomb Raider: Underworld markiert einen Wendepunkt im Texture Mapping, da neben der Diffuse Map nun auch erstmals eine Specular Map zum Einsatz gebracht wird. Diese bezieht sich dabei auf Laras Haut, welche nun nicht nur natürlich schattiert wird

und somit den Körperbau realistischer widerspiegelt, sondern dadurch ist es nun auch möglich, Wassereffekte darzustellen. So glänzt Lara Crofts Haut auch in Situationen, in denen sie nass wird. In dieser Entwicklungsphase ging das Texturing zwar schon einen Schritt weiter als bisher, allerdings fokussierte sich die Konzentration der Entwickler zum derzeitigen Zeitpunkt eher auf modeltechnischen Detailreichtum und Gameplay. So werden Falten und Erhebungen in der Kleidung weiterhin über die Diffuse Map dargestellt. Das Farbkonzept nähert sich dabei wieder dem kühleren, kampfbetonten Eindruck aus *Tomb Raider: Legend* an, übersteigt es sogar zusätzlich. Lara trägt nun nicht mehr nur Waffengürtel im Beinbereich, sondern auch ein sichtbares Umschnallhalter um die Schultern. Der damit kombinierte Rucksack wirkt durch die matt-scharze Farbe und der Andeutung eines Deckel-artigen Verschlusses wie ein Ausrüstungsgegenstand beim Tauchen und passt sich somit dem oftmals im Wasser stattfindenden Gameplay an. Auch die Stiefel gliedern sich erneut in den militärischen Look ein; breite Klettverschlüsse mit metallischem Aufsatz sowie einem mattedem, schwarzem Leder als Material runden den Eindruck ab.

Die Titel der „Legend“ Zeitachse profitieren zunächst einmal erheblich von den neuen Möglichkeiten einerseits der Grafiktechnik – dazu zählen neben dem Fortschritt in der Computergrafik allgemein auch die Nutzung der „Legend“-Engine sowie ab *Tomb Raider: Underworld* die „Underworld“-Engine – als auch andererseits des neuen Entwicklerstudios. Somit sorgen vor allem die Designideen dafür, dass Laras Charakter aus dem höheren Polycount herausgearbeitet werden kann. Daher wurde in dieser Zeitschiene viel mit dem Look des Models experimentiert und unterschiedliche Farb- und Kleidungskonzepte entwickelt. Ihnen allen liegt besonders die Verwendung verschiedener Materialien zu Grunde, die vielleicht noch nicht realistisch gerendert werden und durch mehrere Maps ausgeschmückt sind, jedoch bereits Merkmale aufzeigen, anhand welcher die Oberfläche identifiziert werden kann. Der Einsatz einer ersten Specular Map für die Haut des Models und damit die Darstellung möglicher Effekte auf der Haut läuten dabei die „Survivor“ Zeitachse grafisch ein.

5.3.3 Texturing in der „Survivor“ Zeitachse



Abbildung 24: Zusammenschnitt von Referenzbildern zu Tomb Raider (2013)

Mit Beginn des neuen Handlungsstrangs erfährt die Hauptfigur eine weitere Überarbeitung, mit der sich das Model inklusive Texturing dem fotorealistischen Anspruch annähert. Bereits auf den ersten Blick wird deutlich, dass das Designkonzept von der kurvigen Frau Lara Croft zu einer jungen, zarteren Lara übergegangen ist, deren natürlich schlanke Proportionen stimmig sind. Außerdem wirkt sie weniger erfahren und verletzlicher. *Tomb Raider (2013)* stellt die Anfänge der Protagonistin dar und auf welchem Wege sie zu der tapferen, athletischen und abgehärteten Abenteurerin wurde. Alle Designentscheidungen wurden hinsichtlich dieses Ansatzes getroffen; so wie sich die junge Frau auf eine Expedition begibt, ist sie zwar auf unberührtes, raues Gebiet vorbereitet, auf die tatsächlich eintretenden Geschehnisse jedoch nicht. Nicht nur kämpferische Auseinandersetzungen mit rücksichtslosen Gegnern stellen für Lara eine unbekannte Herausforderung dar, auch zahlreiche ungesicherte und improvisierte Kletterpartien bringen die Protagonistin an den Rand ihrer Kräfte. Das traditionelle Outfit wurde im Zuge dieser neuen Überlegungen ebenfalls angepasst. So weist die Diffuse Map von Laras Kleidung überall Risse und Löcher, von Abstürzen beim Klettern oder Zeugnisse eines Kampfes, auf. Das sonst leuchtend hellblaue Tank Top hat dabei an Sättigung verloren und reicht bis auf Laras Hosenbund, liegt jedoch trotzdem eng am Körper an. Die standardmäßigen kurzen Shorts wurden gegen eine lange, locker sitzende Stoffhose ausgetauscht, welche in einfachen, funktionalen Stiefeln enden. Bei Betrachtung letzterer ist durch die Kombination von Diffuse Map und Specular Map erkennbar, dass es sich um glattes Leder handelt. Etwaige Riemen und Schnallen sind verschwunden, lediglich Schnürsenkel runden den Eindruck von Wanderstiefeln ab. Im Gegensatz zu den sonst so typischen Handschuhen mit Aussparung für die Finger trägt Lara nun eine Art improvisierten Handschutz aus Textilstreifen für das Schießen

mit dem Bogen oder das Klettern an scharfkantigen Felsen. Generell zeugt ihre Ausrüstung von den unerwarteten Ereignissen, sodass Lara einerseits Verletzungen mit Stücken ihrer Kleidung verbindet (wie beispielsweise die zu Beginn der Handlungen zugezogene Verletzung am Oberschenkel durch einen Verband aus weißem Stoff immer noch zu sehen ist) und sie andererseits nur spartanisches Equipment, wie eine kleine Umhängetasche sowie ihrem Funkgerät, bei sich trägt. Aus diesem Grund fehlen auch die traditionellen Waffengürtel und -holster. Zu Beginn der Ereignisse trägt Lara keine einzige Waffe und baut sich, um Wildtiere jagen und essen zu können, aus Zweigen und Textilteilen einen Bogen, welcher die Zwillingspistolen der vorherigen Teile als Markenzeichen für diese Zeitachse ersetzt.

Um diesen Eindruck von Lara Croft authentisch darstellen zu können, ist neben dem Model und der Szenenausleuchtung die Anwendung mehrerer Texture Maps nötig. Die Diffuse Map trägt als Basis den Großteil des Konzeptes bei, indem sie in hoher Auflösung die Beschaffenheit der sich durch das Model abzeichnenden Elemente an Lara sowie das allem zu Grunde liegende kühle blau-braune Farbkonzept abbildet. Um die in diesem Kanal des Materials dargestellten Oberflächen weiterhin zu definieren, erzeugt eine Normal Map Falten, leichte Erhebungen und Absenkungen. Anhand der Art und Weise wie sich die Materialien bei Lichteinfall verhalten, ist die Verwendung einer Specular Map sowie einer Occlusion Map zu erkennen. Besonders deutlich werden die Glanzhighlights auf Laras Haut, welche nicht nur Nässe anzeigt, wenn sie Kontakt mit Wasser hatte, sondern ebenfalls bei anstrengenden Kletterpartien, Sprints, Kämpfen oder auch in der Nähe von Feuer lässt eine schweißnasse Haut auf die Anstrengungen, die sich Lara Croft ausgesetzt sieht, schließen. Somit ist die Specular Map dynamisch, obgleich nicht nur Laras Körper selbst, als auch ihre Kleidung auf die Umwelteinflüsse reagiert. Rutscht sie beispielsweise einen Hang hinunter oder wird verletzt, sodass sie blutet, erscheint ein zusätzlicher *Layer* – Blut oder Schmutz - in der Diffuse Map. Diese Textureffekte erschaffen eine neue Stufe der Menschlichkeit an der Protagonistin, die bisher in der Reihe noch nicht erreicht wurde. Der Spieler kann dadurch Laras Entwicklung nicht nur in dem Moment nachempfinden, wenn das dazu auslösende Ereignis stattfindet, sondern ebenfalls noch danach, sodass sich die Anstrengungen, Improvisationen und Opferungen Laras häufen und am Ende ihre neu gewonnenen Charaktereigenschaften reflektieren.



Abbildung 25: Referenzbilder zu Lara Croft in Rise of the Tomb Raider

In *Rise of the Tomb Raider* wird dieses Texturing-Modell fortgeführt, obgleich aufgrund der Geschichte dieses Titels gewisse Änderungen an Laras Erscheinungsbild vorgenommen werden. Dieser Teil spielt nach den Ereignissen von *Tomb Raider* (2013), sodass die junge Frau nun deutlich abgehärteter sowie erfahrener ist. Am deutlichsten wird dieser Umstand daran erkennbar, dass Lara nun bereits ab Beginn des Spiels eine Waffe besitzt und demzufolge auch wieder Waffengürtel trägt. Diese erscheinen einerseits als Beingürtel mit Holster für eine Pistole sowie für schwerere Kaliber als Umhängegürtel über Schulter und Rücken. Unterstützend für Laras gestärkte Persönlichkeit kommt ebenfalls das Wissen um Feinde, die möglicherweise ebenfalls auf der Suche nach demselben Schatz wie sie sein könnten, hinzu. Diese Erkenntnis lässt Lara nun anstelle eines blauen, auffälligeren Tops ein kühles dunkelbraunes Oberteil tragen, um sich besser tarnen zu können. Die einzigen farblichen Akzente bilden das Gelb des Funkgeräts sowie die Kletteraxt; beides essenzielle Interaktionsgegenstände Laras, welche sie nicht nur benötigt, um ihr Abenteuer zu bestehen, sondern die außerdem einen hohen persönlichen Wert für sie besitzen. Die Axt ist ein Relikt aus ihrem Abenteuer im vorherigen Teil und war ein Geschenk ihres Mentors und Vaterfigur Roth, welcher jedoch in den Folgen der Geschehnisse ums Leben kommt.¹⁵⁴ Das Funkgerät stellt hinzukommend Laras Verbindung mit anderen Menschen dar; während es sie in ihrem ersten Erlebnis vor dem Durchdrehen im Angesicht der Anstrengungen und Schicksale bewahrte, dient es nun weiterhin als Kommunikationsmittel. Es erinnert dadurch immer daran, dass Lara Croft zwar eine starke Person darstellt, die jedoch trotz allem ein fühlender Mensch ist. Besonders weil ihre sonstige Erscheinung in Hin-

¹⁵⁴ IGN (o.J.)

blick auf ihre Reise praktisch, funktional und unauffällig gestaltet ist, sie demnach versucht, unentdeckt ihr Ziel erreichen zu können, stellen diese beiden Gegenstände mehr als nur notwendige Ausrüstung dar.

Das Texturing der beiden Laras der „Survivor“ Zeitschiene fokussiert vor allem die Figur als Forscherin. Ihre Kleidung ist funktional und weniger kämpferisch, als für Klettertouren und Wanderungen durch Ruinen ausgelegt. An vielen Stellen muss Lara improvisieren, um sich den Gegebenheiten der Umwelt sowie der Präsenz ihrer Widersacher anzupassen; Outfit und Ausrüstung gelten dabei als nachvollziehbar für den Spieler. Beides sowie auch Lara selbst korrelieren authentisch mit den Einflüssen, denen sie ausgesetzt ist und zeigt somit verschiedene Facetten von Laras Entwicklung in der Zeitachse. Das Designkonzept zeigt somit sowohl einen spielmechanischen, als auch einen narrativen Ansatz auf, der den Spieler Lara Croft als menschlich reagierende und agierende Hauptfigur empfinden lässt.

5.3.4 Fazit zum Texturing im *Tomb Raider* Franchise



Abbildung 26: Die verschiedenen Lara Crofts im *Tomb Raider* Franchise

Das Texturing evolviert mit zunehmender Zahl an erschienenen Titeln und profitiert dabei von den wachsenden Möglichkeiten in der Computergrafik und Hardwareentwicklung. Das Model Lara Croft ist dabei zunächst in seinen Ursprüngen stark abhängig von der meist männlichen Spielerschaft und dem Ambition, diese innovative Spielidee trotz der ungewöhnlichen Verwendung einer weiblichen Heldin auf dem Markt beliebt zu machen. Mit zunehmendem Interesse an der Reihe, vor allem auch durch weibliche Gamer, entwickelt sich das Konzept der Protagonistin von einer betont

weiblichen Abenteurerin zu einer authentisch aussehenden und handelnden Figur. Das Texturing unterstützt dabei diese Entwicklung und bringt mit den Jahren durch komplexere und detailliertere Diffuse Maps Laras verschiedene Aspekte zum Vorschein. So erscheint sie in manchen Teilen eher als kampferprobt und in anderen mehr als gut vorbereitete Archäologin. Dabei behält sie sich allerdings ein paar herausragende Merkmale, welche, obwohl ihr Gesicht nie ein und dasselbe ist, im Gegensatz zu anderen Spielereihen, sie immer als Lara Croft erkennen lassen. Besonders mit dem Einsatz mehrerer Texture Maps in der letzten Zeitschiene erfährt der Spieler durch narratives Texturing innerhalb eines Titels die Entwicklung, welche Lara über die Jahre der Entstehung des Franchises durchlaufen hat. Und obwohl eine lange Zeit lang kaum von realitätsnahen Texturen die Rede sein konnte, war der realistische Anspruch an das Spiel immer abzusehen, obgleich er sich über die Evolution der Reihe nicht nur erhalten, sondern auch mit jedem Titel gesteigert hat.

6 Fazit

6.1 Zusammenfassende Schlussfolgerungen

Diese Ausführungen lassen erkennen, dass Texture Maps derzeit fest in die Entwicklung digitaler, dreidimensionaler Videospiele etabliert sind. Sie nehmen dabei keine temporäre Stellung als simples Gestaltungsmittel ein, sondern eine konstante Größe mit eigener Maßeinheit, die mit verschiedenen Entwicklungssegmenten des Gesamtprozesses eine Symbiose eingeht.

Das 3D Modeling bietet ihm dabei die Mesh-Topologie als Basisnetz, auf dessen Grundlage die Textur anschließend erstellt werden kann. Allerdings stehen dem Modell dabei nicht unendlich viele Polygone zur Verfügung, da dies eine immens hohe Rechenleistung benötigen würde, die bei Videospielen als Echtzeit-Applikationen nicht tragbar wäre. Sie entsteht vor allem durch den Aufwand, die dreidimensionalen Elemente, die sich im Sichtfeld des Spielers befinden, als zweidimensionales Bild zur Anzeige aufzubereiten. Ein solch rigoroser Verbrauch an Rechenkapazitäten würde dazu führen, dass das Spiel nicht mehr flüssig laufen würde. Diese Qualitätseinbußen müssen unbedingt vermieden werden, um das Spielerlebnis nicht zu gefährden, weshalb der Polycount eines 3D Objekts stetig vom Designer kritisch abgeschätzt werden muss, um so viele Polygone wie möglich einzusparen. Trotzdem stellen Spieler von Videospielen hohe Anforderungen an Ästhetik und Authentizität virtueller Welten, wonach der erforderliche Detailgrad auf andere Weise als durch direktes Modeling generiert werden muss. Texturen erfüllen diesen Anspruch durch ihre Zweidimensionalität; als einfache Bitmap Bilddatei sind sie schneller zu rendern als 3D Objekte und wesentlich Speichersparender. Zudem kann schneller auf sie zugegriffen werden, da sie nur aus Pixeln bestehen, welche mit jeder Art von Software zur Grafikbearbeitung manipuliert werden können. Vor allem aber gibt es eine Vielfalt an Texture Maps, die in der Lage sind, den Detailgrad auf der Oberfläche eines Meshes zu simulieren, der aufgrund des beschränkten Polycounts beim Modeling nicht erzielt werden kann.

Es existieren Mapping Verfahren, welche die Färbung eines Objekts repräsentieren, solche, die die Reaktion des Materials auf direkten und indirekten Lichteinfall widerspiegeln und wieder andere, die Erhebungen und Tiefe auf der Oberfläche darstellen. Jede Map bringt dadurch jeweils verschiedene Aspekte der Beschaffenheit eines Gegenstandes hervor und macht es somit dem Spieler möglich, die Geschichte und den Charakter eines Objektes durch bloße Visualität zu erfahren. Texturen besitzen sogar soweit Potenzial, zusammen mit model-technischen Faktoren wie Proportionen und Shading den Stil eines Spiels maßgeblich zu bestimmen. Die Diffuse Map spielt dabei eine essenzielle Rolle, da ihr aufgrund ihrer Farbinformationen Zugang zur Darstellungs- und Aussagekraft des Gestaltungsmittels „Farbe“ erlaubt ist, wodurch sie den Grundton des Designkonzepts angibt. Fließen realistische Materialien mit in die Textur, wird an dieser Stelle die Basis für eine Tendenz Richtung Realismus gelegt. Im Gegensatz dazu zeigen illustrative Diffuse Maps, erkennbar durch beispielsweise decken-

de Farben, kräftige Konturen oder eine sichtbare Pinselführung, einen stilistischen Ansatz an. Die Ausrichtung des Spiels unterstützt hierbei dessen Narrative, indem es den Wahrnehmungsfokus des Spielers entweder vom Alltäglichen auf das Unbekannte oder vom Fiktiven auf das Natürliche lenkt.

Da jedoch auch Texture Maps Daten darstellen, verbrauchen sie wie jedes andere Element in der Szene ein gewisses Maß an Speicher, da sie eingelesen, verarbeitet und auf dem Bildschirm sichtbar gemacht werden müssen. Aus diesem Grund muss bereits so früh wie möglich im Workflow festgelegt werden, inwiefern das Game Asset im Spiel genutzt wird, um die Anzahl der zu verwendenden Texturen sowie Mapping Typ und Auflösung zu bestimmen. Daher muss als ein erster Schritt über die Priorisierung des Objektes gesprochen werden, wobei zwei Faktoren die Prioritätsstufe bestimmen. Zum einen entscheidet die Häufigkeit des 3D Modells in der Szene über dessen übergeordnete Kategorie – je häufiger ein Asset in der Szene auftaucht, desto mehr Performance wird benötigt, weswegen es in der Priorisierungsebene nach unten rutscht. Zum anderen sorgt die Intensität der Interaktion des Spielers mit dem Objekt für eine höhere Einstufung; in diese Kategorie fallen vor allem der Spielcharakter und Nebenfiguren, die als einzigartige Meshes im Spiel enthalten sind. Je nach Dringlichkeitsstufe müssen die 3D Objekte verschieden gemappt werden, um die Fläche des Modells optimal durch die Textur ausnutzen zu können. Darüber hinaus kann ab dieser Stelle zwar jede Textur dem Mesh zugewiesen werden, es ist jedoch ratsamer, sparsam mit Texture Maps umzugehen und die Menge anhand der Aufmerksamkeit des entsprechenden Assets abzuschätzen, um Speicher zu sparen. Dieses Balancing zwischen Rechenaufwand, Speicherkapazität und grafischer Qualität übernehmen bereits jetzt zum Teil procedural Textures, indem sie die Textur ad hoc über Algorithmen in Echtzeit während des Spiels selbst berechnen und ausgeben. Dieser Ansatz erfordert jedoch höhere Programmierkenntnisse, weshalb momentan eine Zwischenlösung aus manuellem Materialdesign und prozeduraler Generierung genutzt wird. Über die Benutzeroberfläche einer Software wie Allegorithmics Substance Painter können solche Texture Maps erzeugt werden; im Konzeptionsschritt erfährt der Designer dadurch die Möglichkeit, durch die Justierung verschiedener Parameter das Material zusammenzubauen und das Resultat vor allem auf dem Bildschirm direkt einsehen zu können. Die solchen Tools erarbeiteten Texturen profitieren letztlich besonders durch lückenlose Übergänge innerhalb der Textur, eine höhere Wiederverwertbarkeit zu jedem beliebigen Zeitpunkt sowie einem erleichterten Zugriff und Manipulation der Textur selbst.

Texture Maps bilden im Resultat demnach ein Verhältnis aus Gestaltungselement und technischer Raffinesse dar, welches neben Kreativität und dem Verständnis von Materialien ein hohes Maß an Vorbereitung, Kalkül sowie Balance vom Designer erfordert. Bei richtiger Verwendung erlangen Texturen infolgedessen den Status einer essenziellen Schnittstelle zwischen Spiel und Spieler, um ihn über ihre visuelle Informationsgabe narrativ und stilistisch die Geschichte und den Charakter der Spielwelt zu vermitteln. Die Analyse des Texture Mappings anhand des *Tomb Raider* Franchises unterstützt dieses Erkenntnis und zeigt umfassend auf, inwiefern selbst kleine Änderung in den Texturen die Charakteristik einer Spielfigur in eine andere Richtung führt. Zudem wird

deutlich, dass im Laufe seiner Existenz der technische Fortschritt immer mehr Möglichkeiten offerieren konnte, um die Authentizität des Models voranzutreiben. Doch auch wenn die Evolution der Grafikhardware ein höheres Niveau erreicht und dadurch einige Gestaltungsprozesse wie prozedurale Texturen, dem Code überlässt, ist der Designer, der hinter der Produktion von Game Assets steht, immer noch nicht wegzudenken. Die Umsetzung all der Empfindungen, Reize, Ästhetik und Immersion, die durch Texturen geschaffen werden können, kann der Computer zwar effizient leisten, sie aber zu verstehen und nach diesem Wissen zu entscheiden, kann bisher nur der Mensch allein.

6.2 Ausblick der Arbeit und weiterführende Gedanken

Da die Arbeit mit Texturen vorwiegend ein praktisches Unterfangen darstellt, welches jeder Entwickler auf verschiedene Art und Weise angeht, sind analog sowie digital nur geringfügig kompakte Quellen über die angesprochenen Sachverhalte zu finden. Daher sollen die vorangegangenen Untersuchungen und Ergebnisse verschiedenen Interessengruppen erste Auseinandersetzungen mit der Thematik des Texture Mappings aus Design-technischer Perspektive ermöglichen. Die Problematik, mit denen sich besonders junge Entwickler oder Studenten häufig konfrontiert sehen, sind Unmengen von Online-Tutorien über das Erstellen, Bearbeiten und Implementieren von Texture Maps, ohne dass wichtige Hintergrundinformationen vermittelt werden. Solche Lernvideos oder Anleitungen sind zudem nicht in der Lage, einen umfassenden Überblick abzugeben sowie das Texture Mapping in die umliegenden Bereiche der Spieleentwicklung einzuordnen. Diese Arbeit soll ihnen diesen Auftrag abnehmen und somit einen Zugang zum komplexen Umgang mit Texturen bereitstellen. Auch Personen mit der Fachrichtung entsprechendem Lehrauftrag können von der Erörterung dieser Forschungsfrage und den dabei entstandenen Resultaten profitieren. Neben vielleicht sogar neuen Anregungen und Gedanken zur Thematik können Professoren oder Lehrer aufnehmen, inwieweit Auszubildende sich selbstständig mit diesem Thema auseinandersetzen, ausgehend davon Leitfragen rekapitulieren und neu erworbene Informationen aufbereiten könnten. Die Auflistung der verwendeten Quellen kann ihnen hierbei außerdem aufzeigen, wo und durch welche Handlungsträger der Branche sich Interessierte informieren. All diese Erkenntnisse wären möglicherweise hilfreich, um die Inhalte des Lehrplans zu aktualisieren und hinsichtlich der Arbeitsweise und Interessensgebieten der Studierenden zu optimieren.

Insofern, dass diese Arbeit als eine Grundlagenforschung bezüglich des Forschungsgegenstandes ausgelegt ist, kann sie als Basis für weiterreichende Überlegungen dienen. Einerseits könnte beispielsweise der narrative Ansatz von Texture Maps genauer beleuchtet werden, indem anhand von konkreten Beispieltiteln die visuellen Faktoren, die den Spieler durch die Spielwelt leiten, und ihre Wirkungsweise analysiert werden. Die Manipulation des Spielenden durch Farbe, Reizeanordnung und Licht würde in diesem Fall den Hauptuntersuchungsgegenstand repräsentieren, der nicht nur auf Texturen, sondern auch auf den Einsatz der Beleuchtung referiert, wodurch die Symbiose von Texture Mapping und Shading verstärkt aufgegriffen werden könnte. Ziel dieser

Untersuchung könnte es sein, einen Leitfaden für Umgebungsgestaltung in Videospielen aufzubauen, der Kriterien und Umsetzungsvorschläge für die Suggestion des Spielpfades in einem bestimmtem Genre – zum Beispiel Horror – empfiehlt. Andererseits könnte auch etwas spezifischer am technischen Ansatz gearbeitet werden, indem auf die Dependenzen von Modell, Textur und Performanz tief greifender eingegangen wird. Das Verhältnis dieser drei Größen könnte unter der Priorisierung von Games Assets thematisiert werden, sodass sich die Forschungsfrage um die Problematik drehen könnte, in welcher Relation Texture Maps und modelltechnisches Detail wirklich nötig sind, um eine stabile Performanz aufrechterhalten zu können. Denkbare Kriterien wären dabei die Position des 3D Modells im Blickfeld des Spielers, die Anzahl seiner Instanzen in einer Szene, der Interaktionsgrad zwischen Spieler und Objekt oder dessen Integration in die Mechaniken des Spiels. Die Betrachtung dieser Aspekte würde letztlich in einer möglichen Kategorisierung dreidimensionaler Game Assets enden, die konkret aufzeigt, wie die Präferenz für Polycount und Texture Maps eines Assets anhand dieser Richtlinien erfasst werden kann.

Literaturverzeichnis

Literaturquellen

AHEARN, LUKE (2006): 3D Game Textures: Create Professional Game Art Using Photoshop. 1. Auflage, Boca Raton.

BAILEY, MIKE; CUNNIGHAM, STEVE (2016): Graphics Shaders: Theory and Practice. 2., illustrierte Ausgabe, o.O.

BARTEL, STEFANIE (2013): Farben im Webdesign: Symbolik, Farbpsychologie, Gestaltung. illustrierte Ausgabe, Berlin, (X.media.press).

BONANOMI, C., RIZZI, A. (2017): The human visual system described through visual illusions. In: Best, Janet (Hg) (2017): Colour Design. Theories and Applications. 2. Auflage. (The Textile Institute Book Series).

CHOPINE, AMI (2011): 3D ART ESSENTIALS. The Fundamentals of 3D Modeling, Texturing, and Animation. 1. Auflage, Burlington.

DE VIRES, JOEY (2015): Learn OpenGL. An offline transcript of learnopengl.com, 2. Auflage.

EBERT, DAVID S.; MUSGRAVE, F. KENTON; PEACHEY, DARWYN; PERLIN, KEN; WORLEY, STEVEN (2003): Texturing & Modeling: A Procedural Approach. 3. Auflage, San Francisco.

GAGE, JOHN (1999): Color and Meaning: Art, Science, and Symbolism. illustrierte Neuauflage, Singapur.

ORLAND, KYLE; STEINBERG, SCOTT; THOMAS, DAVID (2007): The Videogame Style Guide and Reference Manual. o.O.

PATNODE, JASON (2008): Character Modeling with Maya and ZBrush. Professional Polygonal Modeling Techniques. 1. Auflage, Canada.

HAMMER, NORBERT (2008): Mediendesign für Studium und Beruf: Grundlagenwissen und Entwurfssystematik in Layout, Typografie und Farbgestaltung. illustrierte Ausgabe, Berlin, (X.media.press).

HODGES, ELAINE R. S. (2003): The Guild Handbook of Scientific Illustration. 2. Auflage, Hoboken.

LEE, JOANNA (2016): Learning Unreal Engine Game Development. A step-by-step guide that paves the way for developing fantastic games with Unreal Engine 4. 1. Auflage, Birmingham, (Community Experience Distilled).

MOLLICA, PATTI (2013): Color Theory: An Essential Guide to Color-from Basic Principles to Practical Applications. illustrierte Ausgabe, Irvine, (Artist's Library Series).

NIEVES, ANGEL (2007): Lightwave v9 Texturing. 1. Auflage, Plano.

FLOWMAN, JUSTIN (2016): 3D Game Design with Unreal Engine 4 and Blender. Combine the powerful UE4 with Blender to create visually appealing and comprehensive game environments. 1. Auflage, Birmingham, (Community Experience Distilled).

RUSO, MARIO (2006): Polygonal Modeling: Basic and Advanced Techniques. 1. Auflage, Sudbury.

TAVINOR, GRANT (2009): The Art of Videogames. 1. Auflage, o.O.

THORN, ALAN (2013): Learn Unity for 2D Game Development. 1. Auflage, New York, (Technology in action).

WALLACE, DANIEL (2015): The Art of Rocksteady's BATMAN. Arkham Asylum/Arkham City/Arkham Knight. 1. Auflage, New York.

WATKINS, ADAMS (2011): Creating Games with Unity and Maya. How to Develop Fun and Marketable 3D Games. 1. Auflage, Burlington.

Internetquellen

80.LV (2016): The Artist behind Life is Strange. URL: <https://80.lv/articles/life-is-strange-concept-artist-talks-about-art/> [14.11.2017].

ADOBE SYSTEMS SOFTWARE IRELAND LTD. (2017)[A]: ADOBE PHOTOSHOP CC. URL: http://www.adobe.com/de/products/photoshop.html?sdid=JRSIX&mv=search&s_kwcid=AL!3085!3!235085424711!e!!g!!photoshop&ef_id=Wh-O4gAAADDsBrly:20171130104211:s [30.11.2017].

ADOBE SYSTEMS SOFTWARE IRELAND LTD. (2017)[B]: FAKTEN ZU ADOBE. Lösungen. URL: <http://www.adobe.com/de/about-adobe/fast-facts.html#solutions> [30.11.2017].

ALLEGORITHMIC (2017)[A]: SUBSTANCE DESIGNER: MAKING INCREDIBLE MATERIALS WITH DANIEL THIGER AKA „DETE“. URL: <https://www.allegorithmic.com/blog/substance-designer-making-incredible-materials-daniel-thiger-aka-dete> [15.09.2017].

ALLEGORITHMIC (2017)[B]: SOME GAMES MADE WITH SUBSTANCE. URL: <https://www.allegorithmic.com/games> [15.09.2017].

ALLEGORITHMIC (2017)[C]: SUBSTANCE PAINTER. 3D PAINTING SOFTWARE. URL: <https://www.allegorithmic.com/products/substance-painter> [18.09.2017].

ARCHLINUX (2017): ATI. URL: <https://wiki.archlinux.org/index.php/ATI> [30.11.2017].

AUTODESK INC (2015)[A]: Autodesk Knowledge Network. About Creating 3D Meshes. URL: <https://knowledge.autodesk.com/support/autocad/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/AutoCAD-Core/files/GUID-A6232957-5039-4AB7-8B1D-8FD0AD98F77B-htm.html> [30.11.2017].

AUTODESK INC (2016)[A]: Autodesk Knowledge Network. Hypershade window. URL: <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-252707EC-4AAF-4D3F-9600-804F783652B7-htm.html> [30.11.2017].

AUTODESK INC (2015)[B]: Autodesk Knowledge Network. Normals > Harden Edge. URL: <https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Normals--Harden-Edge-htm.html> [30.11.2017].

AUTODESK INC (2015)[C]: Autodesk Knowledge Network. Normals > Soften Edge. URL: <https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Normals--Soften-Edge-hm.html> [30.11.2017].

AUTODESK INC (2016)[B]: Autodesk Knowledge Network. Polygon normals. URL: <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-9C257D44-924D-4B3F-ADEF-C71FAA98EAB1-hm.html> [11.09.2017].

BALAZS, OLIVER: E-Mail vom 15.09.2017.

CONSTANTINE, THOMAS (O. J.): Video Games: Realism vs. Style in Graphics. URL: <http://www.eecs.umich.edu/courses/eecs487/w07/sa/pdf/tsconst-realism-vs-style.pdf> [13.11.2017].

CORE-DESIGN.COM (o.J.): About. History of the company. URL: <http://www.core-design.com/about.html> [30.11.2017].

CRYSTAL.COM (o.J.): Studio. URL: <https://crystald.com/studio> [30.11.2017].

DAVIS, BEN (2009): CONCEPT DEVELOPMENT FOR GAMES DESIGN. URL: <http://conceptdevelopmentbendavis.blogspot.de/2009/02/what-are-game-assets.html> [01.09.2017].

DEUTSCHER FRANCHISEVERBAND (2017): WAS IST FRANCHISING. URL: <https://www.franchiseverband.com/wissen/franchising-definition/> [30.11.2017].

DONNELLY, WILLIAM (2005): Per-Pixel Displacement Mapping with Distance-Function. URL: http://download.nvidia.com/developer/GPU_Gems_2/GPU_Gems2_ch08.pdf [28.11.2017].

FANDOM (2017)[A]: BioShock Wiki. ADAM. URL: <http://de.bioshock.wikia.com/wiki/ADAM> [31.10.2017].

FANDOM (2017)[B]: BioShock Wiki. Big Daddy. URL: http://de.bioshock.wikia.com/wiki/Big_Daddy [31.10.2017].

FISCHER, MARTIN (2017): Epic-Gründer Sweeney: Der Weg zum Fotorealismus ist steinig. URL: <https://www.heise.de/newsticker/meldung/Epic-Gruender-Sweeney-Der-Weg-zum-Fotorealismus-ist-steinig-3809776.html> [14.11.2017].

GEFORCE (2017): Enabling Ambient Occlusion in Games. URL: <https://www.geforce.com/whats-new/guides/ambient-occlusion#1> [03.11.2017].

GIANT BOMB (2017): Cel-Shading. A style of animation that gives games a more hand drawn look. URL: <https://www.giantbomb.com/cel-shading/3015-173/> [30.11.2017].

GIGA (2016): Was ist GPU? Definition und Erklärung. URL: <http://www.giga.de/extra/gpu/specials/was-ist-gpu-definition-und-erklaerung/> [29.11.2017].

GIMP (o.J.): About. URL: <https://www.gimp.org/about/> [30.11.2017].

IGN (o.J.): Tomb Raider: Definite Edition Wiki Guide. Climbing Axe. URL: http://www.ign.com/wikis/tomb-raider/Climbing_Axe [27.11.2017].

KLAPPENBACH, MICHAEL (2017): Understanding and Optimizing Video Game Frame Rates. URL: <https://www.lifewire.com/optimizing-video-game-frame-rates-811784> [30.11.2017].

LEIFIPHYSIK (2017): Lichtausbreitung. URL: <https://www.leifiphysik.de/optik/lichtausbreitung> [01.11.2017].

LEARN OPENGL (o.J.)[A]: Getting started. Textures. URL: <https://learnopengl.com/#!Getting-started/Textures> [30.11.2017].

LEARN OPENGL (o.J.)[B]: PBR. Theory. URL: <https://learnopengl.com/#!PBR/Theory> [13.11.2017].

LOW, GEK SIONG (2001): Understanding Realism in Computer Games through Phenomenology. Stanford: Stanford University, Facharbeit, Frühling 2001.

MARTEL, BRIAN; NEWMAN, MIKEY; PITCHFORD, RANDY; TIVO, AARON (2010): Borderlands and the 11th Hour Art Style Change. Or: Kids, Don't Try this at Home!. URL: <https://www.gdcvault.com/play/1012480/Borderlands-and-the-11th-Hour> [14.11.2017].

MCCOMBS, SHEA (2017): Intro to Procedural Textures. URL: <http://www.upvector.com/?section=Tutorials&subsection=Intro%20to%20Procedural%20Textures> [15.09.2017].

MEYER, ARNE; NOLAN, NORTH; RUPPEL, ROB; TZENG, FRANK (2016): The Making of Uncharted 4: A Thief's End – Full Movie / Behind the Scenes. URL: <https://www.youtube.com/watch?v=bbx8iFxrVU> [13.11.2017].

MURAVISION (2017): Technology. URL: <https://www.muravision.com/technology.html> [13.11.2017].

NEWZOO (2016): NEWZOO LAUNCHES ONLINE DASHBOARD WITH GAMES MARKET & COMPANY REVENUES. URL: <https://newzoo.com/news/newzoo-launches-online-dashboard-with-games-market-company-revenues/> [11.08.2017].

NEWZOO (2017): THE GLOBAL GAMES MARKET WILL REACH \$108.9 BILLION IN 2017 WITH MOBILE TAKING 42%. URL: <https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/> [11.08.2017].

NOGUER, JEREMIE (2016): THE NEXT FRONTIER OF TEXTURING WORKFLOWS. URL: <https://www.allegorithmic.com/blog/next-frontier-texturing-workflows> [15.09.2017].

NVIDIA (2017): Homepage. URL: <http://www.nvidia.de/page/home.html> [30.11.2017].

PETTIT, NICK (2015)[A]: Asset Workflow for Game Art: Texture Mapping. URL: <http://blog.teamtreehouse.com/asset-workflow-game-art-texture-mapping> [04.09.2017].

PETTIT, NICK (2015)[B]: Understanding Normal Maps. URL: <http://blog.teamtreehouse.com/understanding-normal-maps> [11.09.2017].

PLAYSTATION (2017): PSP™ (PlayStation Portable). Tragbares Gaming für unterwegs. URL: <https://www.playstation.com/de-de/explore/psp/> [29.11.2017].

PLURALSIGHT (2014): Texturing for Games – Maintain a High Level of Detail without Extra Geometry. URL: <https://www.pluralsight.com/blog/film-games/texturing-games-maintain-high-level-detail-without-extra-geometry> [11.08.2017]

POLYCOUNT (2017)[A]: Parallax Map. URL: http://wiki.polycount.com/wiki/Parallax_Map [03.11.2017].

POLYCOUNT (2017)[B]: Topology. URL: <http://wiki.polycount.com/wiki/Topology> [01.12.2017].

RECKER, RUTH-MARIA: Displacement Mapping. Koblenz-Landau: Universität Koblenz-Landau, Studienarbeit, Januar 2006.

ROCKSTEADY LTD. (2017): Homepage. URL: <http://rocksteadyltd.com/#studio> [30.11.2017].

SACCO, DOMINIC (2014): AMD on the future of PC gaming, Mantle and virtual reality. URL: <http://www.pcr-online.biz/news/read/amd-on-the-future-of-pc-gaming-mantle-and-virtual-reality/034732> [11.08.2017].

SCRATCHPIXEL 2.0 (o.J.): Introduction to Ray Tracing: a Simple Method for Creating 3D Images. URL: <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/implementing-the-raytracing-algorithm> [01.12.2017].

SEYMOUR, MIKE (2013): Game environments – Part A: rendering Remember Me. URL: <https://www.fxguide.com/featured/game-environments-parta-remember-me-rendering/> [13.11.2017].

SLANT (2017): What are the best programs to create procedural textures. URL: <https://www.slant.co/topics/4445/~programs-to-create-procedural-textures> [18.09.2017].

SLICK, JUSTIN (2016): 3D Model Components - Vertices, Edges, Polygons & More. Anatomy of a 3D Model. URL: <https://www.lifewire.com/3d-model-components-1952> [01.12.2017].

STERLING, JIM (2009): Borderlands explains its ‘not cel-shaded actually’ art style. URL: <https://www.destructoid.com/borderlands-explains-its-not-cel-shaded-actually-art-style-128892.phtml> [14.11.2017].

STOCKMANN, DANIEL: E-Mail vom 28.11.2017.

STUART, KEITH (2015): Photorealism – the future of video game visuals. URL: <https://www.theguardian.com/technology/2015/feb/12/future-of-video-gaming-visuals-nvidia-rendering> [14.11.2017].

SUN & BLACK CAT (2013): Parallax Occlusion Mapping in GLSL. URL: <http://sunandblackcat.com/tipFullView.php?l=eng&topicid=28> [25.09.2017].

SWEENEY, TIM: Entwickler bei Unreal auf der Gamescom 2017. Mündliche Mitteilung vom 22.08.2017.

SZIRMAY-KALOS, LÁSLÓ; UMENHOFFER, TAMÁS: Displacement Mapping on the GPU: State of the Art. Budapest: Budapest University of Technology, 2006.

TARTACHUK, NATALYA (2012): Those Delicious Texels or ... Dynamic Image-Space Per-Pixel Displacement Mapping with Silhouette Antialiasing via Parallax Occlusion Mapping. URL: https://developer.amd.com/wordpress/media/2012/10/Tatarchuk-ParallaxOcclusionMapping-FINAL_Print.pdf [09.10.2017].

TECHOPEDIA (2017): Gameplay. Definition – What does Gameplay mean?. URL: <https://www.techopedia.com/definition/1911/gameplay> [30.11.2017].

TECHTERMS (2014): File Formats. Bitmap Definition. URL: <https://techterms.com/definition/bitmap> [30.11.2017].

TOMB RAIDER WIKI (o.J.)[A]: Lara Croft. URL: http://tombraider.wikia.com/wiki/Lara_Croft [15.11.2017].

TOMB RAIDER WIKI (o.J.)[B]: Lara Croft (Legend Timeline). URL: [http://tombraider.wikia.com/wiki/Lara_Croft_\(Legend_Timeline\)](http://tombraider.wikia.com/wiki/Lara_Croft_(Legend_Timeline)) [15.11.2017].

TOMB RAIDER WIKI (o.J.)[C]: Lara Croft (Original Timeline). URL: [http://tombraider.wikia.com/wiki/Lara_Croft_\(Original_Timeline\)](http://tombraider.wikia.com/wiki/Lara_Croft_(Original_Timeline)) [15.11.2017].

TOMB RAIDER WIKI (o.J.)[D]: Lara Croft (Survivor Timeline). URL: [http://tombraider.wikia.com/wiki/Lara_Croft_\(Survivor_Timeline\)](http://tombraider.wikia.com/wiki/Lara_Croft_(Survivor_Timeline)) [15.11.2017].

TOMBRAIDER-GAME.DE (o.J.)[A]: CHRONOLOGIE ALLER TOMB RAIDER- & LARA CROFT-GAMES. URL: <http://www.tombraider-game.de/chronologie#tomb-raider> [15.11.2017].

TOMBRAIDER-GAME.DE (o.J.)[B]: LARA CROFT – DATEN & FAKTEN. URL: <http://www.tombraider-game.de/lara-croft> [15.11.2017].

UNITY TECHNOLOGIES (2017)[A]: Unity User Manual (2017.1). Heightmap. URL: <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterHeightMap.html> [25.09.2017].

UNITY TECHNOLOGIES (2017)[B]: Unity User Manual (2017.1). Materials, Shaders & Textures. URL: <https://docs.unity3d.com/Manual/Shaders.html> [09.10.2017].

UNITY TECHNOLOGIES (2017)[C]: Unity User Manual (2017.1). Normal Map (Bump mapping). URL: <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterNormalMap.html> [11.09.2017].

UNITY TECHNOLOGIES (2017)[D]: Unity User Manual (2017.1). Occlusion Map. URL: <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterOcclusionMap.html> [11.09.2017].

UNITY TECHNOLOGIES (2017)[E]: Unity User Manual (2017.1). Textures. URL: <https://docs.unity3d.com/Manual/Textures.html> [04.09.2017].

WHATIS.COM (2005): DEFINITION. code. URL: <http://whatis.techtarget.com/definition/code> [30.11.2017].

ZINK, JASON (2013): A Closer Look At Parallax Occlusion Mapping. URL: <https://www.gamedev.net/articles/programming/graphics/a-closer-look-at-parallax-occlusion-mapping-r3262> [25.09.2017].

Bildquellen

80.LV (2016): Life is Strange: Art of Environmental Storytelling. URL: <https://80.lv/articles/life-is-strange-art-of-environment-stoytelling/> [14.11.2017].

ALLEGORITHMIC DOCUMENTATION (2017): Substance Painter. Interface. URL: <https://support.allegorithmic.com/documentation/display/SPDOC/Interface> [03.11.2017].

ARKHAM WIKI (o.J.): Sionis Steel Mill. URL: http://arkhamcity.wikia.com/wiki/Steel_Mill [02.11.2017].

BAGOGAMES (2016): Venture into the Borderlands – Part 11: Finale. URL: <https://www.bagogames.com/venture-into-the-borderlands-part-11-finale/> [01.11.2017].

DEVIANTART (o.J.): Online Art Network “DeviantArt”. URL: <https://100seedlesspenguins.deviantart.com/art/Bioshock-Adam-Hypo-171605705> [31.10.2017].

ENGADGET (2012): Paul Dini explains absence from next Batman game URL: <https://www.engadget.com/2012/08/06/paul-dini-explains-absence-from-next-batman-game/> [02.11.2017].

FANPOP (o.J.): Uncharted 4: A Thief’s End Nathan and Elena wallpaper and background photos. URL: <http://www.fanpop.com/clubs/uncharted-4-a-thiefs-end/images/40020972/title/nathan-elena-photo> [13.11.2017].

GAME DEBATE (o.J.): Tomb Raider: Legend (PC). URL: http://www.game-debate.com/games/index.php?g_id=124&screenshot=Tomb%20Raider:%20Legend [17.11.2017].

GAME INSIDER (o.J.): Horizon Zero Dawn Review: Guerrilla Games Has Crafted Their Masterpiece. URL: <https://game-insider.com/2017/03/08/horizon-zero-dawn-review-guerrilla-game-has-crafted-their-masterpiece/> [31.10.2017].

GDC VAULT (2010): Borderlands and the 11th Hour. URL: <https://www.gdcvault.com/play/1012480/Borderlands-and-the-11th-Hour> [14.11.2017].

KOTAKU (2017): You Can Play A Bit Of The Original Tomb Raider In Your Browser Right Now. URL: <https://kotaku.com/you-can-play-a-bit-of-the-original-tomb-raider-in-your-1794626757> [16.11.2017].

LARACROFTCOSPLAY.COM (o.J.): Tomb Raider 2013 Clothes and accessories. URL: <http://laracroftcosplay.com/HelpWithOutfits/newtombraiderclothes.html> [27.11.2017].

LARACROFTCOSPLAY.COM (o.J.): Tomb Raider Angel of Darkness Costume breakdown. URL: <http://laracroftcosplay.com/HelpWithOutfits/aodbreakdown.html> [16.11.2017].

LARACROFTCOSPLAY.COM (o.J.): Tomb Raider Legend Costume breakdown. URL: <http://laracroftcosplay.com/HelpWithOutfits/legendbreakdown.html> [17.11.2017].

LARACROFTCOSPLAY.COM (o.J.): Tomb Raider Outfits From All Games. Rise of the Tomb Raider. URL: <http://laracroftcosplay.com/images/jan2016/reference/riseofthetombraiderdeserttanktop.jpg> [27.11.2017].

LARACROFTCOSPLAY.COM (o.J.): Tomb Raider Underworld Costume breakdown. URL: <http://laracroftcosplay.com/HelpWithOutfits/underworldbreakdown.html> [19.11.2017].

METRO (2015): From Sonic The Hedgehog to Lara Croft, these gaming characters have changed a bit over the years. URL: <http://metro.co.uk/2015/02/04/from-sonic-the-hedgehog-to-lara-croft-these-gaming-characters-have-changed-a-bit-over-the-years-5049274/> [16.11.2017].

MOBY GAMES (2008): Tomb Raider II Starring Lara Croft Screenshots (Windows). URL: <http://www.mobygames.com/game/windows/tomb-raider-ii-starring-lara-croft/screenshots/gameShotId,289438/> [16.11.2017].

MOBY GAMES (2008): Tomb Raider: Underworld Screenshots (Windows). URL: <http://www.mobygames.com/game/windows/tomb-raider-underworld/screenshots/gameShotId,378031/> [19.11.2017].

PC GAMER (2015): Dark Souls bonfire found in Just Cause 3. URL: <http://www.pcgamer.com/dark-souls-bonfire-found-in-just-cause-3/> [31.10.2017].

PINTEREST (o.J.): Social Media Platform "Pinterest". URL: <https://www.google.de/url?sa=i&rct=j&q=&esrc=s&source=imgres&cd=&cad=rja&uact=8&ved=0ahUKEwja-7XQvprXAhXRZIAKHSZtDJ0QjRwIBw&url=https%3A%2F%2Fwww.pinterest.com%2F>

atin800%2Frosie-
bd%2F&psig=AOvVaw3QmIxYFPYP1K1uhSwD6TG9&ust=1509526517277854
[31.10.2017].

PINTEREST (o.J.): Social Media Platform "Pinterest". URL:
https://www.google.de/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjO3Ni4m6LXAhUDwxQKHQ3AC_YQjRwIBw&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F629378116645411757%2F&psig=AOvVaw3OKG0HWSJS7K4eXm1gW_ST&ust=1509791556788107 [03.11.2017].

PINTEREST (o.J.): Social Media Platform "Pinterest". URL:
<https://www.pinterest.de/pin/441634307190024731/> [27.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER (2013) SCREENSHOTS. URL:
<http://raidingtheglobe.com/gallery/tomb-raider/screenshots> [27.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER: ANGEL OF DARKNESS OUTFITS. URL:
<http://raidingtheglobe.com/games/classics/tomb-raider-6-the-angel-of-darkness/outfits>
[16.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER: ANNIVERSARY OUTFITS. URL:
<http://raidingtheglobe.com/images/stories/games/tra/outfit/anniversary.jpg> [17.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER: ANNIVERSARY SCREENSHOTS. URL:
http://raidingtheglobe.com/media/com_twojtoolbox/tra_screenshot190.jpg [17.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER: ANNIVERSARY SCREENSHOTS. URL:
http://raidingtheglobe.com/media/com_twojtoolbox/tra_screenshot240.jpg [17.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER: UNDERWORLD SCREENSHOTS. URL:
http://raidingtheglobe.com/media/com_twojtoolbox/tru_screenshot060.jpg [19.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER 1: FEATURING LARA CROFT OUTFITS.
URL: <http://raidingtheglobe.com/games/classics/tomb-raider-1/outfits> [16.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER 2: STARRING LARA CROFT OUTFITS.
URL: <http://raidingtheglobe.com/games/classics/tomb-raider-2/outfits> [16.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER 2: STARRING LARA CROFT SCREENSHOTS. URL: <http://raidingtheglobe.com/games/classics/tomb-raider-2/outfits> [16.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER 4: THE LAST REVELATION OUTFITS. URL: <http://raidingtheglobe.com/games/classics/tomb-raider-4-the-last-revelation/outfits> [16.11.2017].

RAIDING THE GLOBE (o.J.): TOMB RAIDER 4: THE LAST REVELATION SCREENSHOTS. URL: <http://raidingtheglobe.com/games/classics/tomb-raider-4-the-last-revelation/outfits> [16.11.2017].

SEGMENTNEXT (2016): The Evolution of Uncharted Characters, Uncharted 1 to Uncharted 4. URL: <https://segmentnext.com/2016/05/06/evolution-uncharted-characters-uncharted-1-uncharted-4/> [13.11.2017].

THE GAMES CABIN (2015): Microsoft Releases a New Batch of Rise of the Tomb Raider Screenshots. URL: <https://www.thegamescabin.com/microsoft-releases-a-new-batch-of-rise-of-the-tomb-raider-screenshots/> [31.10.2017].

THE INFINITE REVOLUTION (2012): That's it?. URL: <http://theinfiniterevolution.blogspot.de/2012/12/thats-it.html> [14.11.2017].

TOMBRAIDER-GAME.DE (o.J.): TOMB RAIDER LEGEND - SCREENSHOTS XBOX 360. URL: <http://www.tombraider-game.de/tomb-raider-legend/screenshots-xbox-360> [17.11.2017].

WILSON, JOE (2015): PHYSICALLY-BASED RENDERING, AND YOU CAN, TOO!. URL: <https://www.marmoset.co/posts/physically-based-rendering-and-you-can-too/> [13.11.2017].

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Vorname Nachname